Cloud brokering : new value-added services and pricing models

Angel Felipe Diaz Sanchez

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2014-ENST-0028

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité "Informatique et Réseaux"

présentée et soutenue publiquement par

Felipe DIAZ-SANCHEZ

le 10 juin 2014

Cloud brokering :

nouveaux services de valeur ajoutée et politique de prix

Jury

M. Christophe CERIN, Professor, Université Paris 13	Rapporteur
M. Laurent LEFEVRE, Professor, ENS-Lyon, INRIA	Rapporteur
M. Jean-Philippe VASSEUR, Dr. Professeur Associé à Telecom, CISCO-USA	Examinateur
M. Charles LOOMIS, Dr., Laboratoire de l'accélérateur linéaire d'Orsay, CNRS	Examinateur
M. Jean-Pierre LAISNE, Président, société CompatibleOne	Examinateur
M. Maurice GAGNAIRE, Professor, Télécom ParisTech	Directeur de thèse

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

THÈSE

Cloud brokering: new value-added services and pricing models

Abstract:

Cloud brokering is a service paradigm that provides interoperability and portability of applications across multiple Cloud providers. The attractiveness of Cloud brokering relies on the new services and extended computing facilities that enhance or complement those already offered by isolated Cloud providers. These services provide new value to Small and Medium-sized Businesses (SMBs) and large enterprises and make Cloud providers more competitive. Nowadays, at the infrastructure level, Cloud brokers act as an intermediary between the end-users and the Cloud providers. A Cloud broker provides a single point for service consumption in order to avoid vendor lock-in, increase application resilience, provide a unified billing, and simplify governance, procurement and settlement processes across multiple Cloud providers. In the future, Cloud brokers will provide advanced value-added services and will use attractive pricing models to capture potential Cloud consumers. The aim of this thesis is to propose advanced value-added services and a pricing model for Cloud brokers.

Keywords: Cloud Brokering; Value-Added Service; Pricing Model; Cloud Commodity.

Cloud brokering : nouveaux services de valeur ajoutée et politique de prix

Résumé :

Le « *Cloud brokering* » est un paradigme de service qui fournit interopérabilité et portabilité des applications à travers plusieurs fournisseurs de Cloud. Les nouveaux services et capacités étendues qui améliorent ou complètent celles déjà offertes par les fournisseurs de Cloud sont la caractéristique principale des « *Cloud brokers* ». Actuellement, d'un point de vue de l'infrastructure Cloud, les Cloud brokers jouent un rôle d'agents intermédiaires entre les utilisateurs et les fournisseurs, agissant ainsi comme un point commun pour la consommation des services Cloud. Parmi les avantages les plus notables liés à ce point d'accès commun on trouve : l'augmentation de la résilience en allouant l'infrastructure chez de multiples fournisseurs ; la délivrance d'une facturation unifiée ; la simplification des processus de gouvernance ; l'approvisionnement et le règlement à travers de multiples fournisseurs. Dans le futur, les Cloud brokers fourniront des services avancés de valeur ajoutée et vendront des services Cloud en utilisant d'attractives politiques de prix. Le but de cette thèse est de proposer deux services avancés de valeur ajoutée et une politique de prix pour les Cloud brokers.

Mots clés : Courtier Cloud ; Service de valeur ajoutée ; Politique de prix ; Marchandisation du Cloud.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation, objectives and thesis outline

The role of Cloud Brokers in the near future of Cloud computing has been identified by Gartner as a major market trend: "*By 2015, Cloud Brokers will represent the single largest category of growth in Cloud computing, moving from a sub-$1 billion market in 2010 to a composite market counted in the hundreds of billions of dollars.*" [Can12]. This prediction seems to be reinforced by the amount of funding raised by some Cloud brokering companies: *Rightscale* US$47.3m in three rounds[1], *6fusion* US$10m in two rounds, *Cloud Cruiser* US$7.6m in two rounds, *Zimory Systems* US$7.2m in two rounds and *Gravitant* US$3.7m in one round [Fel13]. One of the main reasons, behind this high economic expectation, is the highly heterogeneous current Cloud market constituted by many Cloud providers. Each Cloud provider exhibits different interfaces, pricing models and value-added services. Thereby, to help the end-user cope with such a fragmented ecosystem, Cloud brokers have emerged as an intermediary third-party that provides unified-self service access to multiple Cloud providers. Thus, by being a single point for service consumption, Cloud brokers provide interoperability and portability of applications across multiple Cloud providers. Besides this inherent role, current Cloud brokers provide to Cloud consumers other value-added services, such as follows. Advanced management by using tools beyond the stacks offered by Cloud providers (*e.g.* consolidated billing,

[1] A funding round is a practice by which a company raises money to fund operations, expansion, an acquisition, or some other business purpose.

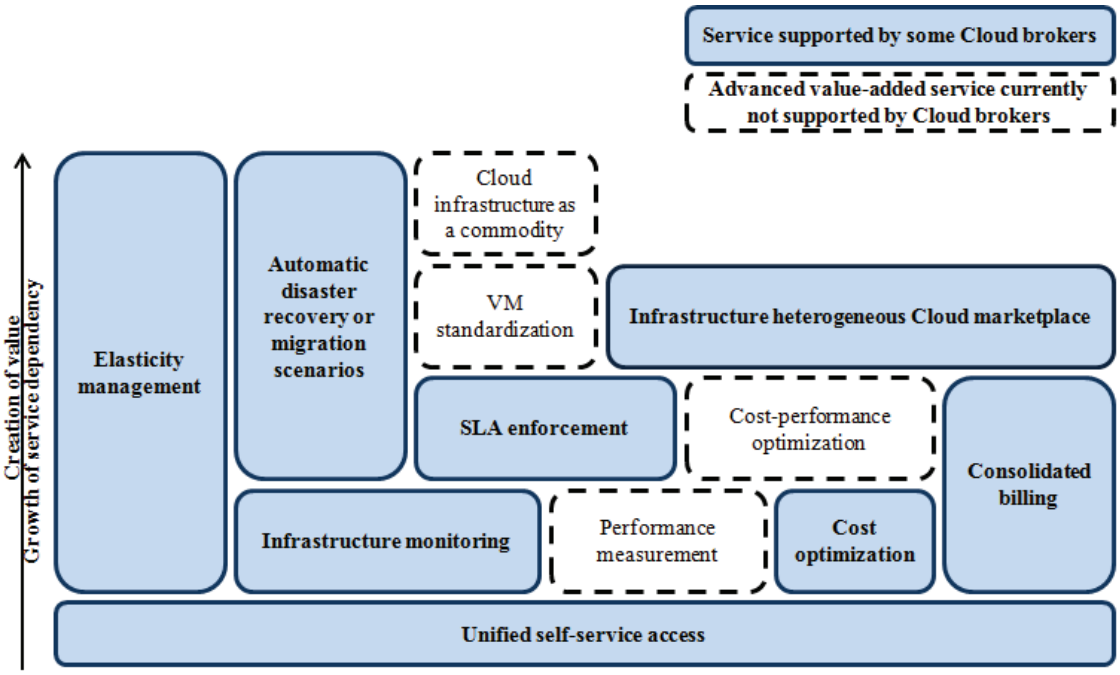

Figure 1.1: Evolution and dependency of value-added services in Cloud brokering

infrastructure monitoring, disaster recovery, SLA enforcement), elasticity management in order to automatically scale up or down infrastructure resources based on the workload and service arbitrage with the aim of taking advantage of two or more Cloud provider offerings (*e.g.* Cost optimization). These services can be overlayed, enabling new Cloud computing scenarios such as Cloud bursting or Cloud marketplaces (Figure 1.1). These new scenarios may be beneficial for both end-users and Cloud providers. In the case of Cloud bursting, end-users have the possibility to extend their computing facilities by moving the development of applications or the non-mission-critical applications to public Clouds. In the case of a Cloud marketplace scenario, end-users have access to multiple Cloud providers through a single interface, while Cloud providers may sell spare infrastructure capacity.

Cloud brokers are expected to drive creation of value through advanced value-added services enabling new Cloud computing scenarios. The price of Cloud computing resources varies around 20% between Cloud providers, while the performance differences between Cloud providers remain unknown or less studied [Fel13]. Due to the fact that Cloud brokers are able to deploy a workload in any Cloud provider, the measurement of performance of Cloud providers and the placement of Cloud resources based on a cost-performance relationship may be in the future value-added services supported by Cloud brokers. Moreover, the commoditization of infrastructure resources will increase the Cloud adoption by simplifying the purchase of Cloud computing resources. Being Cloud computing resources traded like any other commodity (*e.g.* wheat, oil, iron) will flatten the current fragmented Cloud market. This opens the door to new pricing models

in which Cloud brokers not only will act as intermediaries but also as liquidity providers, negotiating volume discounts from Cloud providers and guaranteeing resource availability to end-users.

Service arbitrage enables advanced services in Cloud brokering by taking advantage of two or more Cloud provider offers. This allows Cloud brokers to simplify the vast number of offers by categorizing the features and benefits of each Cloud provider in order to match consumer needs with an ideal set of Cloud providers. In the first part of this thesis entitled "Value-added services in Cloud brokering", it is carried out a comprehensive state of the art on Cloud performance evaluations and placement in Cloud brokering (Chapter 2). Then, it is proposed a method to calculate Cloud performance through a single figure of merit based on the mapping of the physical features of a VM to their respective performance capacities (Chapter 3). Finally, it is proposed an exact placement approach for optimizing the distribution of Cloud infrastructure across multiple providers (Chapter 4). Parameters such as price, VM configuration, VM performance, network latency and availability are considered for that purpose.

Nowadays, *pay-as-you-go* and *reserved* pricing dominate the way consumers acquire Cloud resources from legacy Cloud providers at the infrastructure level. However, the introduction of Cloud brokers may induce the commoditization of Cloud infrastructures. Facing such an evolution, new pricing models are necessary to capture potential consumers or untapped market segments. The second part of this thesis entitled "A new pricing model for Cloud brokering" focuses on the design of a pricing model for Cloud brokering, called *pay-as-you-book* (Chapter 5). *Pay-as-you-book* is based on two types of information. The first type consists of the forecast of users' job requests. The second one consists of the ability of Cloud brokers to take advantage of such advanced reservations. With this aim in view, a study comparing three resource allocation policies under *pay-as-you-book* is carried out.

The aim of this thesis is to contribute to the design of new value-added services and pricing models for Cloud brokering. The majority of the investigations and original results presented in this manuscript have been achieved and obtained in the context of the CompatibleOne [COn] research project supported by the French Ministry of Industry. Its objective was to demonstrate the feasibility of a Cloud brokering intermediation platform integrating and adapting the various software solutions proposed by the industrial and academic partners of the project. This platform provides a single point for service consumption in order to avoid vendor lock-in. This thesis has three objectives:

- The first one is to propose a single figure of merit of Cloud VMs performance based on the application profile.
- The second one is to propose an exact approach for allocation of VMs across multiple Cloud providers based on different optimization criteria.

- The third one is to describe a pricing model for Cloud brokering, called *pay-as-you-book*.

1.2 Contributions of this thesis

The contribution of this doctoral research can be itemized as the following:

- A method to calculate a figure of merit of VM Cloud performance. The originality of this figure of merit is to offer a single value to express VM Cloud performance that is based on the type of application to be deployed. Thus, end-users may in a straightforward manner compare and select the best Cloud provider in which to deploy an application.
- The formulation of Mixed-Integer Linear Programming for placement of VMs across multiple Cloud providers. The originality of this approach is in associating the heterogeneity of Cloud providers' offers with their respective performance. This approach may be applied to the optimization of cost, performance, cost-performance and disaster recovery scenarios.
- The description of pay-as-you-book, a pricing model between pay-as-you-go and subscription. Pay-as-you-book consists of paying and reserving time-slots of VMs in advance without a fixed fee to subscribe to the service and without a long-term commitment, avoiding vendor lock-in, while obtaining lower prices than in pay-as-you-go. Pay-as-you-book may be applied in scenarios with predictable workloads. Through simulations, it has been shown why a model such as pay-as-you-book is not convenient for Cloud providers. However, Cloud brokers reselling Cloud infrastructure may create attracting service offerings based on pay-as-you-book.

1.3 Publications

This dissertation consists of an overview of the following conference publications:

1. F. Díaz-Sánchez, S. Al Zahr, M. Gagnaire, J-P. Laisné, J. Marshall. "CompatibleOne: Bringing Cloud as a Commodity". *IEEE International Conference on Cloud Engineering (IC2E)*, Boston, US, May. 2014.
2. F. Díaz-Sánchez, S. Al Zahr, and M. Gagnaire. "An Exact Placement Approach for Optimizing Cost and Recovery Time under Faulty Multi-Cloud Environments". *IEEE CloudCom Conference*, Bristol, UK, Dec. 2013.
3. F. Díaz Sánchez, E. Doumith, S. Al Zahr and M. Gagnaire. "An Economic Agent Maximizing Cloud Providers Revenues Under Pay-as-you-Book Pricing Model".

Conference on the Economics of Grids, Clouds, Systems, and Services (GECON), Berlin, Germany, Nov. 2012.

4. F. Díaz, E. Doumith and M. Gagnaire. "Impact of Resource over-Reservation (ROR) and Dropping Policies on Cloud Resource Allocation". *IEEE CloudCom Conference*, Athens, Greece, Nov. 2011.

5. F. Díaz-Sánchez, M. Gagnaire, J. Marshall and J-P. Laisné. "COSCHED: A Scheduling Agent Maximizing Cloud Broker's Revenues under the CompatibleOne Architecture". *The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA-13)*, Melbourne, Australia, Jul. 2013.

Part I

Value-added services in Cloud brokering

Chapter 2

State of the art: Cloud performance and placement in cloud brokering

Contents

2.1 Introduction

The growing number of Cloud computing services increases the interest of consumers in comparing these services in order to choose those best adapted to their needs. This chapter focuses on the performance issues related to Cloud provider evaluation and on the role of Cloud brokers in the automatic optimization of resource allocation across multiple Cloud providers. This chapter is structured as follows. A survey of the current studies related to Cloud performance evaluation appears in Section 2.2. The motivations and challenges behind the evaluation of Cloud provider performance are described. Section 2.3 presents the state of the art on placement in Cloud brokering. The studies are classified into

two categories: placement based on non-functional requirements and application-aware placement.

2.2 Cloud performance evaluation

2.2.1 Motivations and challenges

The current Cloud computing landscape hinders a straightforward comparison of Cloud provider service offerings. In the case of computing resources, this is mainly due to the heterogeneity of VM configurations and prices. On the one hand, traditional Cloud providers such as Amazon, Rackspace and WindowsAzure sell fixed-size VMs. These VM configurations vary from one Cloud provider to another, therefore it is not possible to find the same VM configuration at two Cloud providers. On the other hand, new Cloud providers in an effort to attract consumers, look to differentiate their services through technology by allowing consumers to configure at freely the size of the computing resources to be purchased.

VM performance evaluation adds another layer of complexity to the comparison of Cloud providers. Firstly, consumers have little knowledge and control over the infrastructure hosting their applications. Due to the virtualization of hardware used in Cloud computing, Cloud providers may use resource sharing practices (*e.g.* processor sharing, memory overcommit, throttling or under-provisioned network [HZKD11]) that degrade the performance of a Cloud application. Secondly, Cloud provider's data centers are equipped with hundred of thousands of servers with different qualities of hardware and software. Thereby, the evaluation of performance, across all the data centers of multiple Cloud providers, implies a trade-off between thoroughness, time and cost of the evaluation [LYKZ10]. Thirdly, Cloud providers may continually upgrade or extend their hardware and software infrastructures, and new commercial services and technologies may gradually enter the market [LZO+13]. Therefore, performance evaluations become quickly out of date and the tools for performance measurement must be continuously re-designed. Finally, there are no Cloud-specific benchmarks to evaluate all VM features [IPE12]. However, traditional benchmarks can partially satisfy the requirements for Cloud performance evaluation.

Cloud performance evaluation would be beneficial for both consumers and Cloud providers [LYKZ10]. Consumers testing their applications across multiple Cloud providers can choose the Cloud provider that represents the best performance-cost trade-off. Also, performance evaluations can serve as a recommendation of the performance of a particular system [HZKD11] or can give to consumers technical arguments to put pressure on Cloud providers to use best practices [IPE12]. A provider may identify its market positioning in order to improve its services or to adjust its prices [LYKZ10].

2.2.2 Studies related to Cloud providers performance evaluation

An exhaustive study about the academic approaches of commercial Cloud services evaluation has been carried out by the Australian National University [LZO+13]. A Systematic Literature Review (SLR) was the methodology employed to collect the relevant data to investigate the Cloud services evaluation. As a result, 82 relevant Cloud service evaluation studies were identified. The key findings of this study represent a state-of-practice when evaluating Cloud services and are as follows:

- 50% of the relevant studies investigated applying Cloud computing to scientific issues, while only 16% of the studies focused on the evaluation of business applications in the Cloud.
- 21 Cloud services over 9 Cloud providers were identified. 70% of the relevant studies evaluated Cloud services provided by Amazon Web Services (AWS).
- Three main aspects and their properties for Cloud services evaluation have been investigated: performance, economics and security, performance being the most studied aspect (78 studies).
- There is no consensus regarding the definition and the usage context of metrics. Some metrics with the same name were used for different purposes, some metrics with different names were essentially the same. The study identified more than 500 metrics including duplications.
- There is a lack of effective metrics vis-à-vis elasticity and security aspects in Cloud computing. Therefore, it is hard to quantify these apects.
- There is not a single or a small set of benchmarks that provides a holistic evaluation of Cloud services. The SLR identified around 90 different benchmarks in the selected studies of Cloud services evaluation. These benchmarks can be grouped in three main categories: application, synthetic and micro-benchmarks, as explained below.
- 25 basic setup scenarios for constructing complete Cloud service evaluation experiments have been identified and classified.
- The Cloud service evaluation is getting more and more attention from the research community. The number of relevant studies was 17 times larger in 2011 (34 studies) than in 2007 (2 studies).

Cloud performance evaluation is done by running application benchmarks, synthetic benchmarks or micro-benchmarks in single or multiple Cloud providers. Application benchmarks correspond to real-world software that provides an overall view of the performance of a specific application. Synthetic benchmarks simulate application behavior by imposing a workload on the system. Similarly, micro-benchmarks impose a workload with the aim of measuring hardware-specific VM features. Since there are no Cloud-specific

benchmarks, Cloud performance has been measured through widely used benchmarks such as TPC-W (a transactional web e-Commerce benchmark) [LOCZ12], HPCC (a software suite consisting of 7 basic benchmarks) [SASA+11, IOY+11a, HZKD11], NPB (set of parallel benchmarks to evaluate the performance of parallel supercomputers) [MVML11, HZKD11] or common measurement tools such as *ping* or *iperf* [SDQR10, BK10]. Also, specific benchmarks have been developed to measure Cloud performance of CPU, memory, disk and network [ADWC10, HLM+10] further the VM provisioning or deprovisioning time [SDQR10, IOY+11a, MH12]. Details about the studies related to Cloud providers performance evaluation are presented in Table 2.1.

Recent studies tend to clarify confusing concepts, inaccurate terms, as well as to unify the metrics used by previous Cloud performance evaluation studies. Li *et al.* propose a taxonomy of performance for evaluating commercial Cloud services [LOCZ12] and potential approaches to bring a holistic impression of Cloud services performance through a single figure of merit [LOZC13].

Table 2.1: Studies related to Cloud providers performance evaluation

Study	Type of benchmark	Applications or Suite/Benchmarks	Property	Metric
Stanchev [Sta09]	Synthetic	WSTest	Overall performance	Transactions per second
Yigitbasi et al. [YIEO09]	Application	Modified Grenchmark	Overall performance	Queue waiting time (s)
		Modified Grenchmark	Overall performance	Response time (s)
	Micro	Benchmark developed by authors	Elasticity	VM adquisition and release (s)
Dejun *et al.* [DPC09]	Application	CPU-intensive web	CPU	Duration of operation (ms)
		Database read-intensive	Disk	Duration of operation (ms)
		Database write-intensive	Disk	Duration of operation (ms)
Baun and Kunze [BK10]	Application	Compilation Linux Kernel	CPU	Duration (s)
	Micro	Bonnie++	Disk	KBps
		Bonnie++	Disk	Number of file operations/s
		iperf	Network	Transfer rate in KBps
		ping	Network	RTT (ms)
Alhamad *et al.* [ADWC10]	Application	Java application	Network	Response time (ms)

Continued on next page

Table 2.1: (Continued)

Study	Type of benchmark	Applications or Suite/Benchmarks	Property	Metric
El-Khamra *et al.* [EKKJP10]	Application	EnKF-based matching	CPU	Total execution time (s) per number of processor cores
Hill *et al.* [HLM+10]	Synthetic	TPC-E	Overall performance	Average transaction time (s)
	Micro	Not specified	Elasticity	VM adquisition and release (s)
		Not specified	Elasticity	Time per web role action (s)
		Not specified	Network	RTT (ms)
		Not specified	Network	Bandwidth (MBps)
Schad *et al.* [SDQR10]	Micro	Ubench	CPU	Ubench score
		Ubench	Memory	Ubench score
		Bonnie++	Disk	Total execution time (KB/s)
		iperf	Network	TCP throughput (Mbps) intra-datacenter network
		Application developed by authors	Elasticity	VM adquisition (s)
He *et al.* [HZKD11]	Application	CSFV	CPU	Total execution time (s)
	Synthetic	NPB	CPU	Total execution time (s)
		HPCC/HPL	CPU	GFLOPS
	Micro	iperf	Network	Message Latency (s)
		iperf	Network	TCP throughput (bps) intra-datacenter network
Moreno-Vozmediano *et al.* [MVML11]	Synthetic	GridNPB/ED	Overall performance	Throughput (jobs/s)
		NAS/NGB	Overall performance	Throughput (jobs/s)
Vöckler *et al.* [VJD+11]	Application	Scientific workflow application	Overall performance	Jobs/s
Ostermann *et al.* [IOY+11b] extended to 4 providers in [IOY+11a]	Synthetic	HPCC/HPL	CPU	GFLOPS
		HPCC/RandomAccess	Network	MBps
	Micro	lmbech/all	Many	Many
		HPCC/DGEMM	CPU	GFLOPS
		CacheBench	Memory	MBps
		HPCC/STREAM	Memory	GBps
		HPCC/b_{eff}	Memory	GBps
		Bonnie	Disk	MBps

Continued on next page

Table 2.1: (Continued)

Study	Type of benchmark	Applications or Suite/Benchmarks	Property	Metric
		Benchmark developed by authors	Elasticity	Duration (s)
Phillips *et al.* [PEP11]	Application	Gromacs, FFmpeg, Blender	Many	Total execution time (s)
	Synthetic	TORCH/Dhrystone	CPU	Total execution time (s)
		TORCH/Spectral	CPU	Total execution time (s)
		TORCH/Particle	CPU	Total execution time (s)
Salah *et al.* [SASA+11]	Micro	Simplex	CPU	Total execution time (ms)
		HPCC/STREAM	Memory	MBps
		FIO	Disk	KBps
Lenk *et al.* [LML+11]	Micro	Phoronix/crafty,dcraw	CPU	Test duration (s), MFLOPS
Li *et al.* [LOCZ12]	Application	TPC-W	Overall performance	Page generation time(s)
	Synthetic	Modified SPECjvm2008	CPU	Finishing time of a CPU-intensive task (s)
		Modified SPECjvm2008	CPU	Finishing time of a memory intensive task (s)
		Modified SPECjvm2008	Memory	Finishing time of a disk I/O intensive task (s)
		Modified SPECjvm2008	Disk	Finishing time of a CPU-intensive task (s)
	Micro	iperf	Network	TCP throughput (Mbps) in intra- and inter-datacenter network
		ping	Network	RTT (ms)
Mao et Humphrey [MH12]	Micro	Benchmark developed by authors	Elasticity	VM acquisition and release (s)

2.2.3 Cloud VM characterization

According to the studies of Cloud provider performance evaluation presented in the previous Section, a Cloud VM can be represented by a set of criteria and a set of capacities (Figure 2.1). The criteria set is composed of the VM physical properties (*i.e.* communication, computation, memory and storage) and of Cloud service related features (*i.e.* availability, reliability, scalability and variability). The set of capacities corresponds to

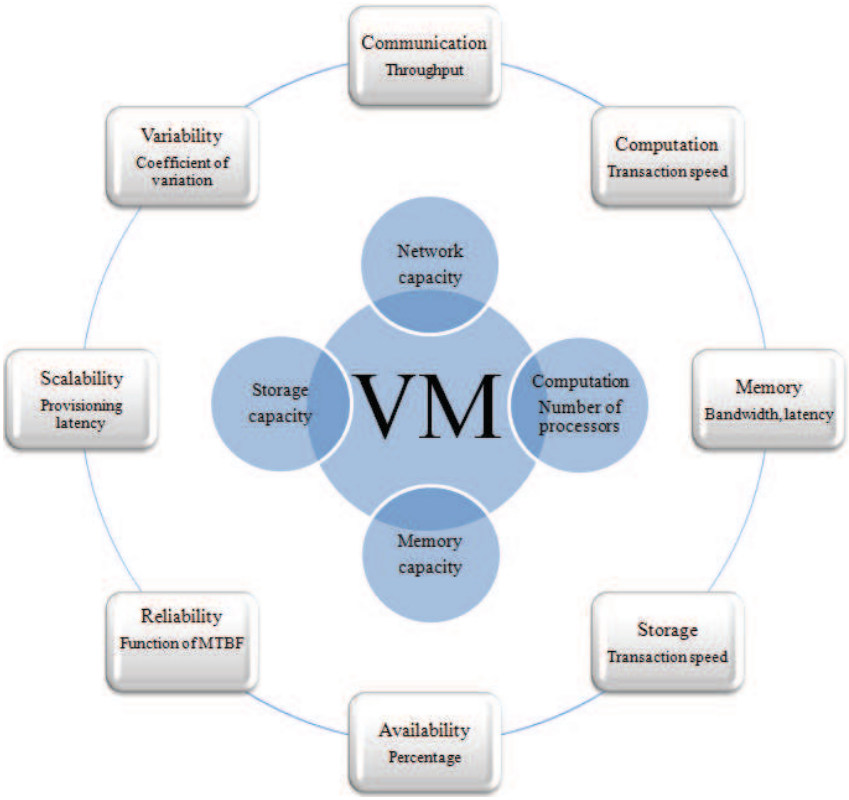

Figure 2.1: VM characterization. The inner circle represents the physical properties of a VM. The outer circle presents the performance criteria and examples of their capacities.

the metrics used to describe the performance of the criteria. Both criteria and capacities are described below.

Criteria

- *Communication* is defined as the property of transferring data between two entities through a network. Three types of communications can be distinguished: intra- and inter-datacenter network, and wide-area network [LYKZ10]. Intra-datacenter network refers to the communication of two VMs belonging to the same datacenter, while inter-datacenter network corresponds to the communication between two VMs located in different datacenters but belonging to the same Cloud provider. Wide-area network refers to the communication between a VM located in a datacenter and an external host on the Internet.
- *Computation* refers to the physical property of processing data. In the case of a VM, computation corresponds to the evaluation of the virtual CPU.
- *Memory* corresponds to the physical property of storing data on a temporary basis. Both RAM memory and cache are considered in this category.

- *Storage* refers to the physical property of storing data on a permanent basis, until the data is removed or the service is suspended by the end-user.

- *Availability* is defined as the percentage of time an end-user can access the Cloud service (Equation 2.1). For a given interval of time, it is calculated as a ratio of the uptime of the Cloud service to the total time of the interval, usually on a yearly basis.

$$\text{Availability} = \frac{\text{total uptime}}{\text{total time of the interval}} \tag{2.1}$$

- *Reliability:* In the literature, the definition of reliability varies depending on different contexts or perspectives. Here, reliability refers to the property of a Cloud service to perform its function for a specified period of time (Equation 2.2). It is defined based on the previous failures experienced by users and the promised Mean Time Between Failures (MTBF) by the Cloud provider [GVB13].

$$\text{Reliability} = \left(1 - \frac{\text{number of users experiencing a failure}}{\text{number of users}}\right) \times \text{MTBF} \tag{2.2}$$

  Thus, if a Cloud provider promises a MTBF of 8760 hours (one failure per year) and 20% of his clients experienced a failure in an interval less than promised by the Cloud provider. The reliability performed by the Cloud provider, according to our definition is 7008 hours (or 9 months and 22 days).

- *Scalability (also known as elasticity)* is the ability at which the application capacity can be adapted to the demand of end-users [ILFL12]. Two types of scalability can be distinguished: *Horizontal* [VRMB11, WCC12] and *Vertical* [DTM11, YF12]. The former refers to the provisioning of multiple instances of the Cloud service (*e.g.* deploy new VMs). The latter implies to add more resources to a current Cloud service (*e.g.* add dynamically more processors or storage to a VM).

- *Variability (also known as stability)* refers to the variation of performance of a Cloud service. Unlike the availability and the reliability that are either provided by the Cloud provider or that can be easily calculated, variability depends on the values of the capacities (as explained below). Therefore, variability can be considered as a derived capacity. Several metrics have been employed as a metric to evaluate variability [LOZC12]. Here, we have used the Coefficient of Variation (CV), which is defined as the ratio of the standard deviation to the mean (Equation 2.3). The CV is useful for comparison between data sets with different units (as it is the case of most of the benchmarks), since it allows to compare the degree of variation from one data set to another.

$$\text{CV} = \frac{1}{\bar{x}} \cdot \sqrt{\frac{1}{N-1} \cdot \sum_{i=1}^{N} (x_i - \bar{x})^2} \tag{2.3}$$

Here N is the number of measurements; $x_1, ..., x_N$ are the measured results; and $\bar{x}$ is the mean of those measurements.

Capacities

The presented capacities have been defined by Li *et al.* in [LOCZ12].

- *Transaction speed* defines how fast transactions (*e.g* job execution, read/write operation) can be processed.
- *Data throughput (Bandwidth)* is considered as the amount of data processed by any physical property in a given period of time.
- *Latency* includes all the time-related capacities of a Cloud service.
- *Other* consists of dimensionless metrics (*i.e.* availability, CV) or single metrics such as the reliability.

2.3 Placement in Cloud brokering

The placement or resource allocation in Cloud brokering refers to the mechanisms to distribute infrastructure resources across multiple Cloud providers based on end-users' needs and constraints. The optimization goal in placement is to select a single or a set of Cloud providers to optimally deploy a service based on an optimization criteria, for example cost optimization or performance optimization. Placement mechanisms can be classified into *non-functional requirements-based placement* and *application-aware placement*. The non-functional requirements placement corresponds to the allocation of Cloud infrastructure based on the match of both Cloud provider resources and end-user requirements. The application-aware placement is based on the constraints that guarantee a Quality of Service (QoS) of the application running on top of the infrastructure.

2.3.1 Non-functional requirements-based placement

Placement studies based on non-functional requirements consider performance of Cloud providers and/or dynamic pricing scenarios[1]. In the literature, we have identified two Cloud brokering placement scenarios: static and dynamic. Static placement assumes that changes within the Cloud environment never happen. Dynamic placement addresses the issue of how to reconfigure Cloud resources optimally, adapting them to new situations

[1]Another non-functional requirements, out of the scope of this dissertation, cover end-users limiting the set of placement solutions due to political and legislative considerations. For example, end-users could avoid placing data either outside or inside a given region (*e.g.* the EU Data Protection Directive which regulates the processing and free movement of personal data within the European Union).

when conditions change (*e.g.* Cloud provider outage, new VM prices, and *etc.*). The approaches described below are based on exact models (*e.g.* binary integer programming formulation).

Static placement

Tordsson *et al.* [TMMVL12] propose an architecture for Cloud brokering and a placement algorithm based on the performance of the GridNPB/ED benchmark and the price of resources. An end-user may constrain resource deployment by specifying the type and number of VMs to be deployed and the percentage of VMs located within each Cloud provider. Chaisiri *et al.* [CLN09] propose an optimal VM placement across multiple Cloud providers that considers both reserved and on-demand provisioning plans. However, including the reservation plan implies not only a long commitment in exchange of lower prices regarding on-demand service provisioning but also raises new issues in case of underprovisioning or overprovisioning of IaaS resources. On the one hand, in the underprovisioning scenario, the demand can be fully met through on-demand resources at a higher cost. On the other hand, in the overprovisioning scenario, questions arise such as: who (the end-user or the Cloud broker) is going to pay for the unutilized IaaS resources?

The usefulness of Cloud brokering placement for fully-decoupled or loosely-coupled applications is studied by Van den Bossche *et al.* [VdBVB10] and Moreno-Vozmediano *et al.* [MVML11]. Both approaches improve the cost-effectiveness of the deployment and consider an on-demand provisioning plan and a hybrid IaaS Cloud architecture. Van den Bossche *et al.* [VdBVB10] propose a cost-optimal placement for preemptible but non-provider-migratable batch workloads with a strict completion deadline. The workloads are characterized by memory, CPU and data transmission requirements. The problem is tackled by Linear programming. Moreno-Vozmediano *et al.* [MVML11] evaluate the scenario of deploying a computing cluster on top of a multi-Cloud infrastructure for solving loosely-coupled Many-Task Computing (MTC) applications. The goal is to improve the cost-effectiveness of the deployment, or to implement high-availability strategies. This approach is evaluated through a low scale testbed including a local data-center and three different public Cloud sites. This testbed is complemented with simulations that include a larger number of resources.

Dynamic placement

Lucas-Simarro *et al.* [LSMVML11] propose a VM placement algorithm with the goal of minimizing the costs for end-users in a dynamic pricing environment. The Cloud broker transfers clients' infrastructure from one Cloud provider to another based on price fluctuations. The algorithm calculates possible future prices based on the average Cloud

provider's price and its price trend. In order to guarantee the performance of the applications running on top of the IaaS resources, the placement decisions are constrained by: the maximum and minimum number of VMs to reallocate in each placement and a load balancing requirement that indicates the percentage of resources to maintain within each Cloud provider. In this approach, the placement problem is limited to one VM configuration. Lucas-Simarro [LSMVML12] extends this work to multiple VM configurations and addresses the problem of performance optimization. Performance optimization consists in maximizing the performance of the deployed resources by choosing the VMs with the best performance in terms of hardware resources (hard disk, memory, CPU). A drawback of this approach is that VM performance measures should be provided by end-users after testing all VM configurations within each Cloud provider.

A more complex model that not only involves cost-optimization but also copes with changes in the Cloud environment through VM migration is proposed by Tordsson *et al.* [LTE11]. In this model, the time for VM migration is approximated by the time required to shut down a VM within one Cloud provider and start a new VM with the same configurations within another.

Chaisiri *et al.* [CLN12] propose an optimal Cloud resource provisioning algorithm minimizing the cost of resource provisioning for a certain period given the uncertainty for demand and price. The optimal decision calculated by the Cloud broker is based on end-users' demands and Cloud providers' prices. This allows the Cloud broker to adjust the number of resources acquired in advance under reservation and the number of resources to be acquired under on-demand provisioning, taking into account that reserved VMs are generally cheaper than on-demand ones. This approach tackles the underprovisioning and overprovisioning problem. Chaisiri addresses this problem through stochastic integer programming.

2.3.2 Application aware placement

The application-aware placement dynamically scales up or down resources across multiple Cloud providers' infrastructures under QoS constraints specific to the application. In the case of tightly-coupled applications with low delay or strong communication requirements, the placement process should guarantee a single-cloud deployment [GB12]. On the other hand, in the case of fully-decoupled[2] or loosely-coupled applications, the placement process may take advantage of the heterogeneity of Cloud providers' offers to deliver a cost-effective solution that guarantees the performance of the application [RCL09, VdBVB10]. In the case of interactive applications (*e.g.* on-line gaming), user experience relies on network bandwidth and on the latency caused by geographical distances [GB12].

[2]Applications are fully-decoupled when the jobs that form the application have no precedence constraints, and can be executed in parallel.

Therefore, these kinds of applications should be treated near the geographical location of their origin to achieve lower latency and higher throughput.

The importance of Cloud brokering for telecommunication services is highlighted by Carella G. *et al.* [CMCS12]. In this approach, the Cloud broker enhances his placement mechanisms based on: real-time data on network performance, QoS requirements and Cloud providers' prices. The goal is to provide to telecommunication service operators a minimum QoS to satisfy customer's requirements by monitoring the deployed services. This approach is evaluated in a testbed composed of a Cloud broker and an IP Multimedia Subsystem (IMS) deployment. The cost-effective placement of Web 2.0 applications with high-availability and fault-tolerance requirements across multiple Cloud providers is proposed by Frincu *et al.* [FC11]. In this approach, authors consider applications consisting of several components and connectors (C/Cs). C/Cs are reallocated by making a snapshot, stopping the execution of each C/C, moving the snapshot to a new VM and starting the C/C from the snapshot. A Cloud broker architecture with the intelligence to react to changes in business processes by changing the Cloud configuration across multiple Cloud providers is described by Grivas *et al.* [GKW10].

The placement of services with different QoS and service provisioning requirements for risk assessment services and e-learning education applications is tackled by Quarati *et al.* [QCGD13]. The goal is to maximize user satisfaction and broker's revenues by reducing energy costs, through energy saving mechanisms. For this, the Cloud broker allocates IaaS resources to the public or private Cloud based on end-user's QoS expectations and the workload of the private resources. This approach was evaluated through a discrete event simulator.

2.4 Conclusion

In this chapter, it has been surveyed research work tackling the problem of Cloud performance evaluation and placement in Cloud brokering. A shortcoming in the current approaches to Cloud performance evaluation is the absence of a single figure of merit that provides a straightforward comparison of Cloud providers. Regarding the problem of placement in Cloud brokering, the surveyed studies assume that Cloud providers offer the same type of VM configurations. This assumption is not true for all the cases; VM configurations may vary from one Cloud provider to another. For some cases, even a VM offered by a Cloud provider in one location, may not exist in another location belonging to the same Cloud provider. These issues are tackled in the next two chapters.

Chapter 3

Towards a figure of merit of Cloud performance

Contents

3.1 Introduction

Let's imagine creating a figure of merit for automobiles and that the most expensive Mercedes-Benz has the highest figure of merit. Does this mean everyone should buy that particular car? What if you want to tow a trailer? In Cloud computing this also happens,

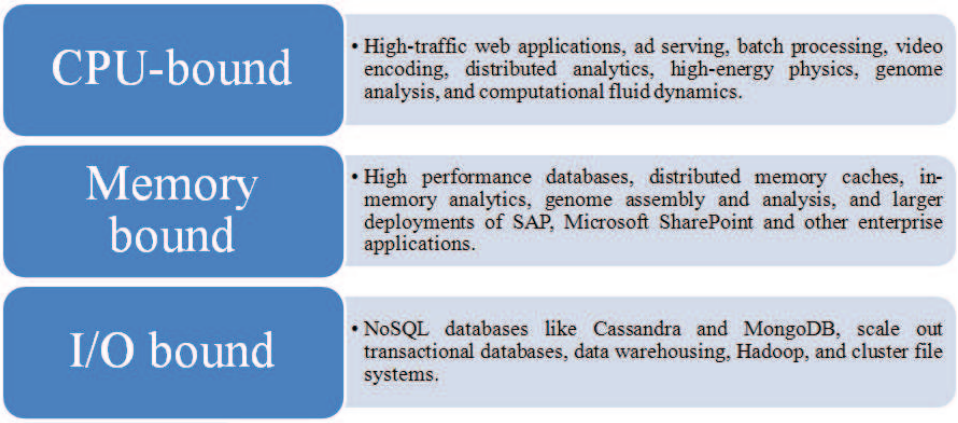

Figure 3.1: Examples of types of applications[a]

[a]Source: http://aws.amazon.com/ec2/instance-types/

although we may be able to find a single value to represent Cloud performance, it does not mean that this value will be useful for all type of applications. According to the physical property that limits performance, applications can be classified as CPU-bound, memory bound or I/O bound (Figure 3.1). Therefore, the application profile must be taken into account in the calculation of a single figure of merit.

Currently, the information given by Cloud providers allows a simple but inaccurate comparison between providers. End-users can choose a Cloud provider by comparing quantitatively different VM offers (*e.g.* number of cores per dollar, memory or storage capacity per dollar). Using this simple approach, end-users can select the Cloud provider that offers the largest quantity of resources at the lowest price. However, this only makes sense in the unreal scenario in which Cloud providers have qualitatively homogeneous resources. Another more precise alternative for comparing Cloud providers, is to evaluate the performance of an application across multiple Cloud providers and to choose the set of Cloud providers with the best performance-cost ratio [LSMVML12]. This approach is technically feasible through a Cloud broker but requires a time-consuming and highly expensive task: the application must be evaluated in each VM configuration offered by each Cloud provider. Moreover, the performance estimation becomes increasingly inaccurate as the Cloud providers upgrade their infrastructures (Section 2.2.1). In contrast with the methods described above, a figure of merit of Cloud performance based on the application profile, can serve as a general guide for the performance of a particular system configuration and provide a straightforward comparison of Cloud providers for a given type of application. In this chapter, a figure of merit of Cloud performance is calculated, by running benchmarks in advance across multiple Cloud providers and by obtaining a composed metric based on the benchmarks' results.

The motivation behind a figure of merit in Cloud brokering is twofold. Firstly, the performance of the Cloud providers can be measured in advance with consistent results for

a period of time. This is due to the fact that the time to run benchmarks is much shorter than the upgrade cycle of Cloud infrastructures. Secondly, a Cloud broker, granting access to multiple Cloud infrastructures as a trusted third-party, may provide up-to-date evaluations of Cloud provider performance. Cloud brokers may automatically deploy benchmarks and process the results. Thereby, a Cloud broker can easily automate the process for calculating a figure of merit.

This chapter is organized as follows. In Section 3.2, the methodology and the experimental setup used in our Cloud evaluation is described. This is followed by the evaluation of the provisioning time and the evaluation of criteria such as computation, memory, storage and variability for different types of VMs and Cloud providers. In Section 3.3, two approaches to calculate a single figure of merit of Cloud performance are presented. Finally, Section 3.4 presents a case study for a CPU-intensive application. In this case study, we use real performance results to compute a figure of merit with three different methods.

3.2 Performance evaluation

3.2.1 Evaluation methodology

There is a lack of standardized methodology for Cloud performance evaluation through benchmarks (*cf.* Section 2.2.2). Here, we describe the methodology employed in this work to measure Cloud performance. This methodology is composed of five main steps (Figure 3.2):

1. *Define scenarios:* the stakeholders (*i.e.* Cloud providers to be evaluated) are identified, as well as the features related to the Cloud services such as VM configurations and datacenter locations.

2. *Identify Benchmarks:* selection of suitable benchmarks according to the scenario formulated in the previous step. If we want to evaluate the performance of a specific application, this step is omitted. Evaluation-related issues such as the number of benchmark repetitions and the type of workload are defined in this step [LOCZ12].

3. *Run tests:* the resources are acquired on the selected Cloud providers' locations. Then, benchmarks are deployed into the chosen VM configurations. At the end of this step, the results are collected and the resources are released.

4. *Process results:* the results are treated and synthesized. For example, by calculating a figure of merit of Cloud performance or by generating a graphical representation that summarizes the main results, aspects and trends.

Figure 3.2: Evaluation methodology

5. *Analyze results:* In this final step, comments or recommendations are formulated based on the results.

3.2.2 Experimental setup

Our initial idea was to create an Operative System (OS) image consisting of all the scripts and benchmarks necessary to measure Cloud performance. This image would be uploaded and deployed in every Cloud. Thus, we would guarantee the same workload conditions for every Cloud provider. However, the Cloud providers, covered in this study, present issues that hinder or completely prevent VM import. In some cases, Cloud providers only support the import of VMs generated via licensed software (*e.g.* Amazon only support the import of images generated with VMware vSphere Client). In other cases, the import of VM images is only supported for some but not for all the image formats (*e.g.* Cloudsigma only supports import of VMs in *RAW* format). Finally, particularly in recent-emerged Cloud providers, the import of VM images is not supported at all. For these reasons, it has been opted to build a VM image from the images already offered by Cloud providers. The experimental setup presented here consists of three phases: image setup, running benchmarks and processing benchmark results. These phases occur once the accounts have been created in every Cloud provider to be evaluated and the payment details have been registered. More issues related with this evaluation of performance are presented in Appendix A.1.

The image setup is as follows. First, a VM via web interface or command line is created. During the VM creation, the OS system is chosen. In this setup, Linux CentOS 6.X for a 64 bits processor architecture, an OS supported by the majority of Cloud providers, has been chosen. Once the VM has been created, the OS is updated and a *ssh* server is installed to enable a secure remote control of the VM. Then, the scripts, benchmarks and tools necessary to evaluate Cloud performance are installed and configured. The scripts have been developed in Python. The execution permissions of the */etc/rc.local* file have been modified and changed, in order to automatically trigger the benchmarks once the VM is turned-on. The *phoronix-test-suite* [pho] has been selected as the framework to deploy benchmarks due to its widely set of supported benchmarks (more than 350). For the transmission of benchmark results, *s3cmd*, a command line tool for using the Amazon S3 service, has been used. The image containing all the scripts, benchmarks and tools necessary to evaluate Cloud performance has been called *ceilo*(Figure 3.3).

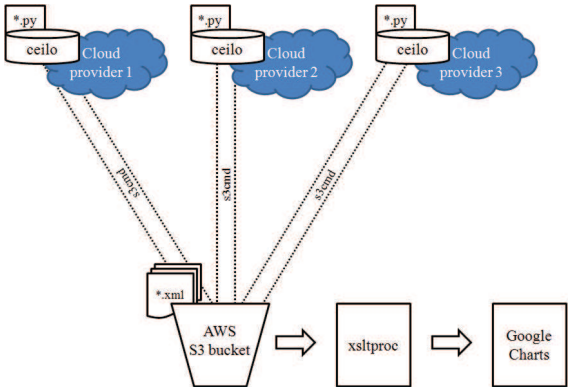

Figure 3.3: Experimental setup

Id	Cloud provider
ARU	ArubaCloud
AWS	Amazon Web Services
CLO	Cloudsigma
JOY	Joyent
LUN	Lunacloud
PRO	Profitbricks
RAC	Rackspace
WIN	WindowsAzure

Table 3.1: Evaluated Cloud providers

As explained above, the benchmarks are triggered automatically and sequentially once the VM is turned-on. Once all the benchmarks have been executed, the results are sent to an Amazon S3 bucket and the VMs are automatically turned-off. Benchmark results correspond to XML files. Thus, result files are parsed with *xsltproc* [xsl], a command line tool for applying XSLT stylesheets to XML documents; and values such as the average, the variance, the standard deviation and the coefficient of variation are calculated. Finally, for some of the results, we run a script to automatically generate graphical representations of the results with *google charts*.

In this study, the performance of 37 different VM types across 8 Cloud providers with data centers in Europe has been measured (Table 3.1). According to the pre-configured VMs offered by some Cloud providers and in order to compare VMs with similar capacities, we have defined five different VM sizes (*xs*, *s*, *l*, *m* and *xl*) (Table A.1). This classification has been based on the number of virtual CPUs (vCPUs) and the RAM memory size.

The performance evaluation presented here is based on 6 benchmarks. These benchmarks measure the computation, memory and storage capacities (Table 3.2). Depending on the operation implemented by the benchmark, the magnitude of the results can be classified as: Lower is Better (LB) or Higher is Better (HB). LB means that the lower the value, the better the system to execute a given benchmark. Inversely, HB means that the

Criteria	Capacity	Benchmark	Metric	Type
Computation	Transaction speed	7zip [zip]	MIPS	HB
		C-ray [cra]	seconds	LB
Memory	Data throughput	Stream [Str]	MB/s	HB
		CacheBench [Cac]	MB/s	HB
Storage	Transaction speed	Threaded I/O Tester [TIO]	MB/s	HB
		Iozone [ioz]	MB/s	HB

Table 3.2: Benchmarks

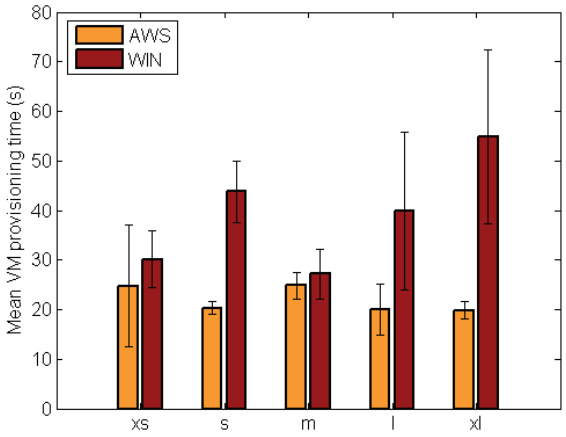

Figure 3.4: Average VM provisioning time for Windowsazure and Amazon

higher the value, the better the system to execute a given benchmark. The provisioning time has been measured with our own scripts. In this section, the mean values and the standard deviation of the obtained results have been plotted. An analysis related to the benchmark duration is presented in Appendix A.3.

3.2.3 Provisioning time

The provisioning time (or scaling latency) is defined as the time taken by a Cloud provider to allocate a new VM once the end-user requests it [LYKZ10]. The provisioning time corresponds to the sum of the time that takes to a Cloud provider to power-on a VM (VM provisioning time) and the boot time of the OS, defined as the time between when the VM has been powered-on and the VM is ready to be used. The provisioning time has a direct impact in the scalability of a Cloud application, particularly in peak load scenarios, where the deployment of Cloud infrastructure must follow the workload variations and the VMs must be ready to be used as soon as possible.

The VM provisioning time for every VM size of WindowsAzure and Amazon has been measured (Figure 3.4). In general, WindowsAzure has a higher provisioning time and a larger standard deviation than Amazon. The VM provisioning time for Amazon stays under $25s$ while for WindowsAzure it is consistently over $25s$.

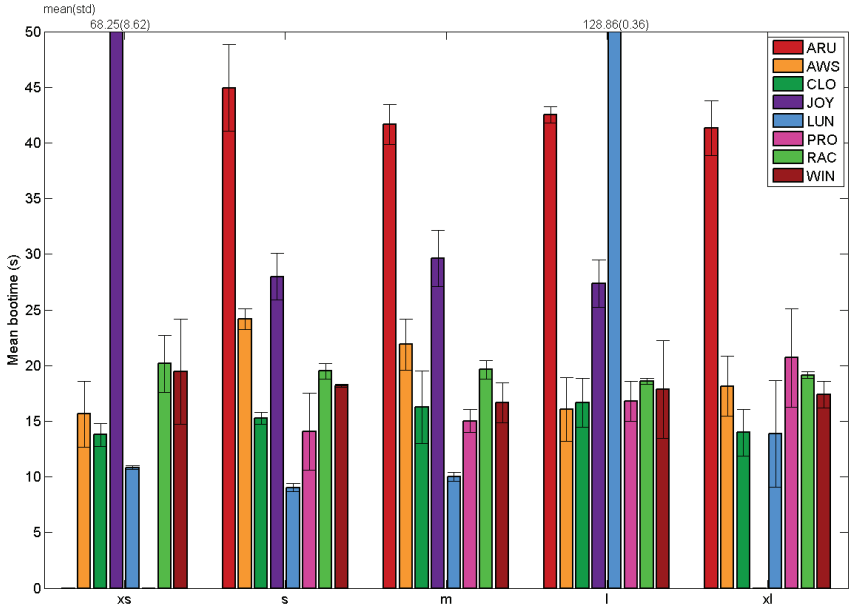

Figure 3.5: Average boot time

The boot time for every VM size of every Cloud provider has been also measured (Figure 3.5). In 5 out of 8 evaluated Cloud providers, the VM boot time varies in the range 10-25s and is independent of the VM size. In the case of Joyent, the boot time is inversely proportional to the VM size. Arubacloud presents a VM boot time that varies in the range 40-50s and it is independent of the VM size. Lunacloud presents the lowest boot time values for *xs*-, *s*-, *m*- and *xl*-VM sizes. Lunacloud's *l*-size VMs present the highest boot time among all the evaluated Cloud providers. There is not a logical explanation to this fact, from the data we have collected. We found, Lunacloud's *l*-size VMs shared the same processors family (Intel Xeon E5-2620) with the other Lunacloud's VM sizes. Thus, the processor brand probably is not the reason for these boot time differences. Unfortunately, we do not have either additional results or information about the Lunacloud's underlying infrastructure to determine the reasons behind this high boot time.

3.2.4 Computation benchmarks performance

The transaction speed has been measured with *7-zip* and *C-ray* benchmarks (Figure 3.6). 7zip is an application to compress files. The benchmark consists of compressing a file with random data and measuring the number of CPU instructions executed during the compression. C-ray measures floating point CPU performance. By default, the benchmark uses only a small amount of data, such that on most systems the CPU does not have to access the RAM to run the benchmark. In our performance evaluation, C-ray was set up to measure the time to render an image with a resolution of 800x600 pixels. Therefore

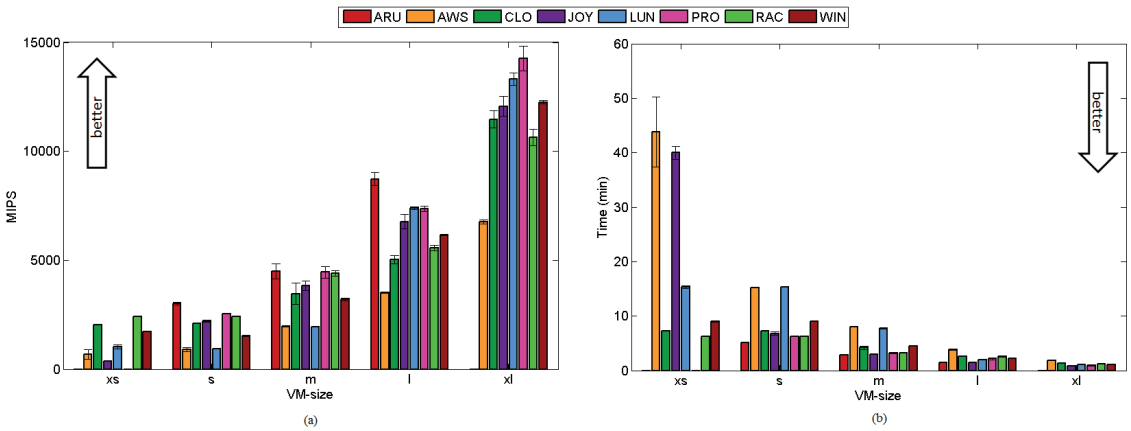

Figure 3.6: Performance of computation benchmarks (a) 7-zip results (HB) (b) C-ray results (LB)

unlike for 7-zip in C-ray, the lower results are better.

3.2.5 Memory benchmarks performance

RAM and cache memory bandwidth have been measured with the *Stream* and *Cachebench* benchmarks, respectively (Figure 3.7). Stream is a simple synthetic benchmark program that measures sustainable memory bandwidth and the corresponding computation rate for simple vector kernels. In our performance evaluation, Stream was set up to measure the memory bandwidth through the *copy* and *add* operations. The copy operation consists of fetching two values from memory and updating the value of one of these fetched values with the other. The add operation fetches three values from memory and updates one of the fetched values with the sum of the other two fetched values. Cachebench is a benchmark designed to evaluate the performance of the cache memory present on a system. In this performance evaluation, CacheBench was set up to measure the cache memory bandwidth through *read* and *write* operations. In general, CacheBench results show the writing speed is around 60%-80% faster than the reading speed (Figure 3.7e).

3.2.6 Storage benchmarks performance

The storage bandwidth has been measured with the *Iozone* and *Threaded I/O Tester (TIO)* benchmarks (Figure 3.8). Iozone is a filesystem benchmark tool. The benchmark generates and measures a variety of file operations. In this performance evaluation, Iozone was used to measure the transaction speed for reading and writing a file of 2GB. Similarly, the write and read speeds have been measured with TIO for a 64MB file by using 16 threads. For the small VM sizes (*xs* and *s*), the read and write speed are comparable for both benchmarks. For the *m*-, *l*- and *xl*-VM sizes, the read speed is at least ten times

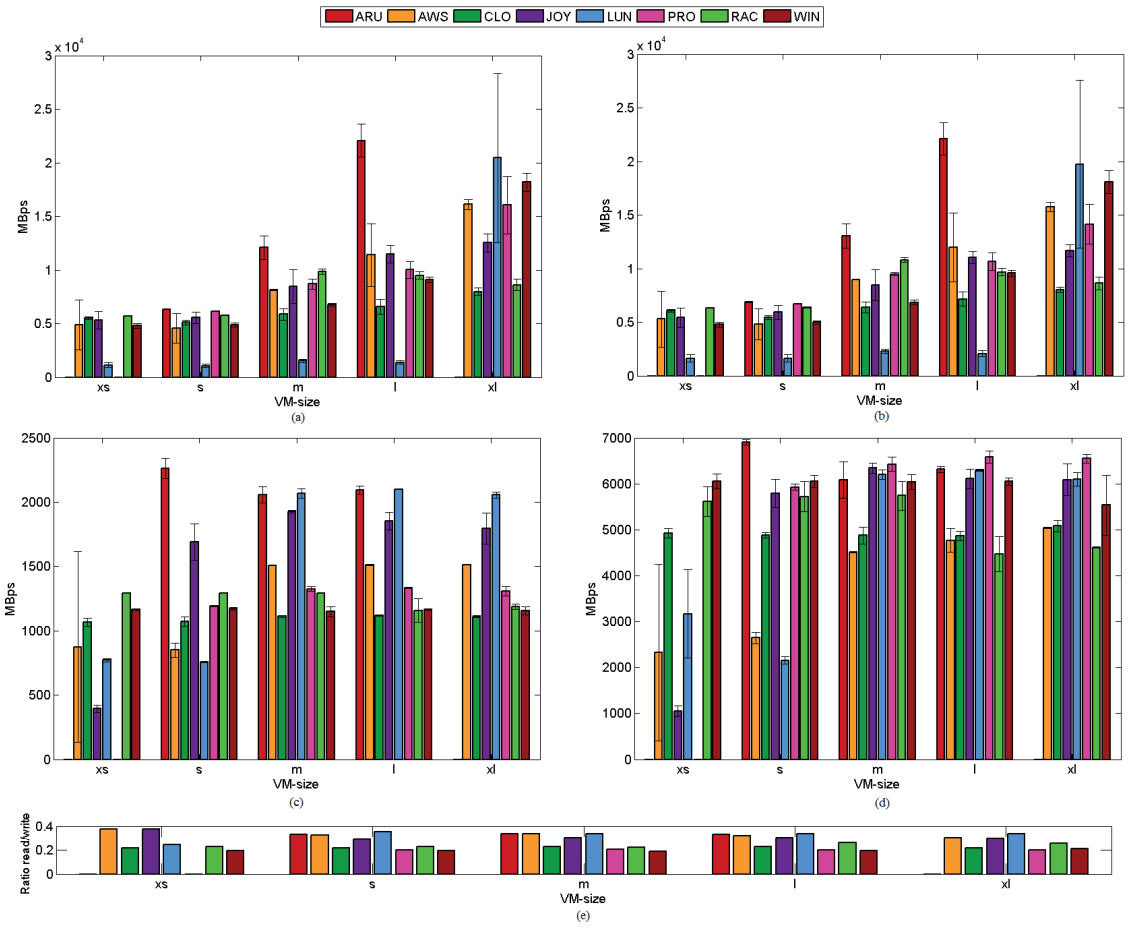

Figure 3.7: Performance of memory benchmarks. (a) Stream results for copy operation (b) Stream results for add operation (c) CacheBench results for read operation (d) CacheBench results for write operation (e) Ratio CacheBench read/write speed.

faster than the write speed (Figure 3.8c). Iozone's read/write ratio is bigger than 150 for Amazon (Figure 3.8f).

3.2.7 Variability

The variability of VMs has also been studied. For this, a single value of variability has been calculated by averaging the Coefficient of Variation (CV) of all the benchmark results. The distribution of variability (Figure 3.9) shows that 70.3% of the evaluated VMs have a variability less or equal to 10%. Since physical servers host many VMs at the same time, one should expect that the bigger the VM size, the lower the variability, and viceversa. However, results show that even small VM sizes present low variability values. The percentage of VMs with a variability between 40% and 45% corresponds to the *xs*-VM size of AWS. One possible explanation to this fact is that the number of processor of the AWS's *xs*-VMs is not constant, providing spiky CPU resources [aws].

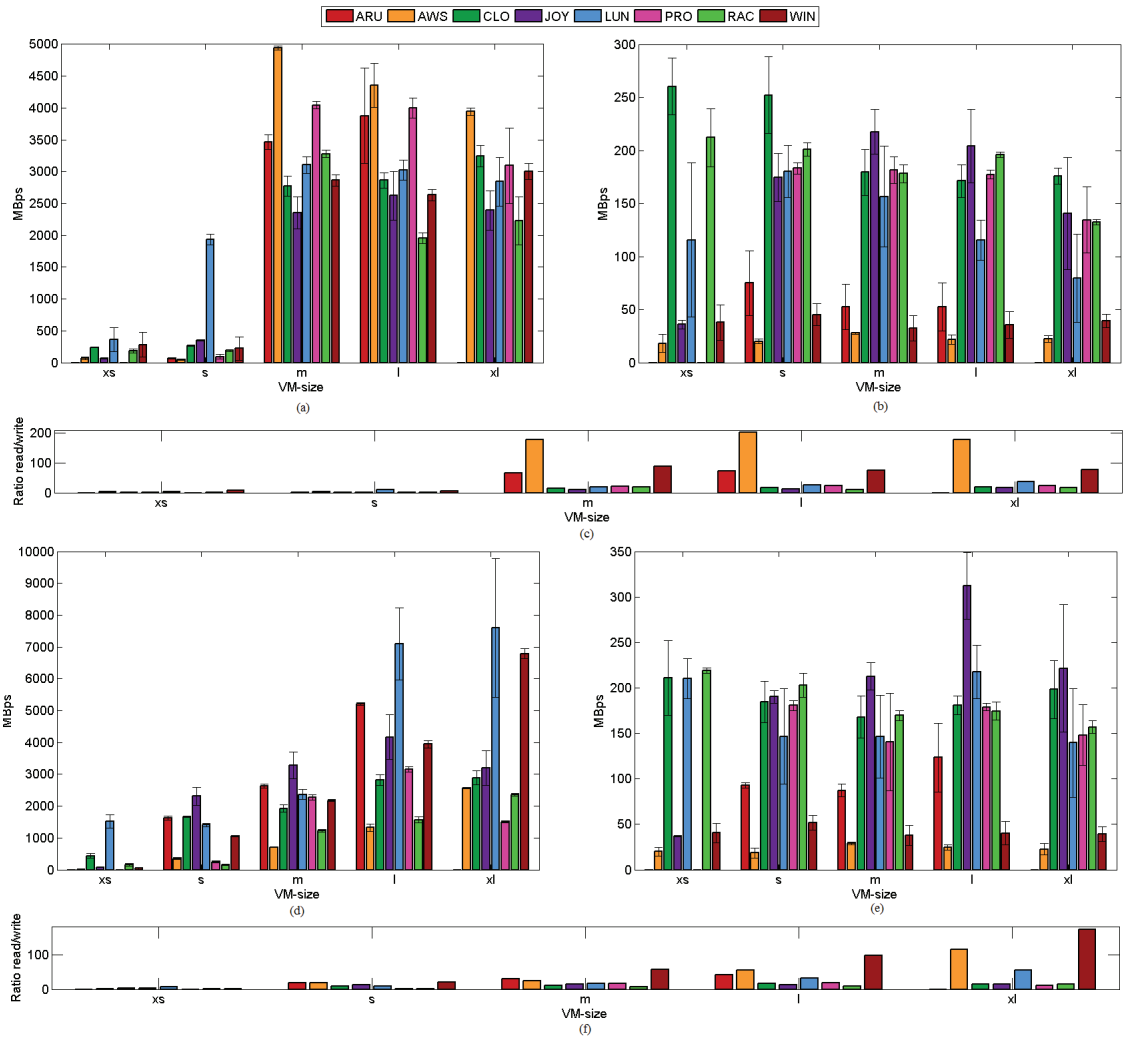

Figure 3.8: Performance of memory benchmarks. (a) Iozone read speed (b) Iozone write speed (c) Ratio Iozone read/write speed (d) TIO read speed (e) TIO write speed (f) Ratio TIO read/write speed.

3.3 Figure of merit of VM Cloud performance

No benchmark offers a holistic view through a single score of Cloud performance. Instead, benchmarks have their own specific metrics and magnitudes to express results. This heterogeneity in results prevents a straightforward, simple calculation of an absolute figure of merit of VM performance. Moreover, even in the case of benchmarks sharing the same units to express results, it is incorrect to directly add values from different benchmarks. The reason is that the magnitudes of values can differ significantly, for example read and write cache results differ by a magnitude of three. Therefore, it is not an easy task to choose a Cloud provider based on individual benchmark results. In this section, some methods to calculate a figure of merit of VM performance to allow a simple Cloud provider

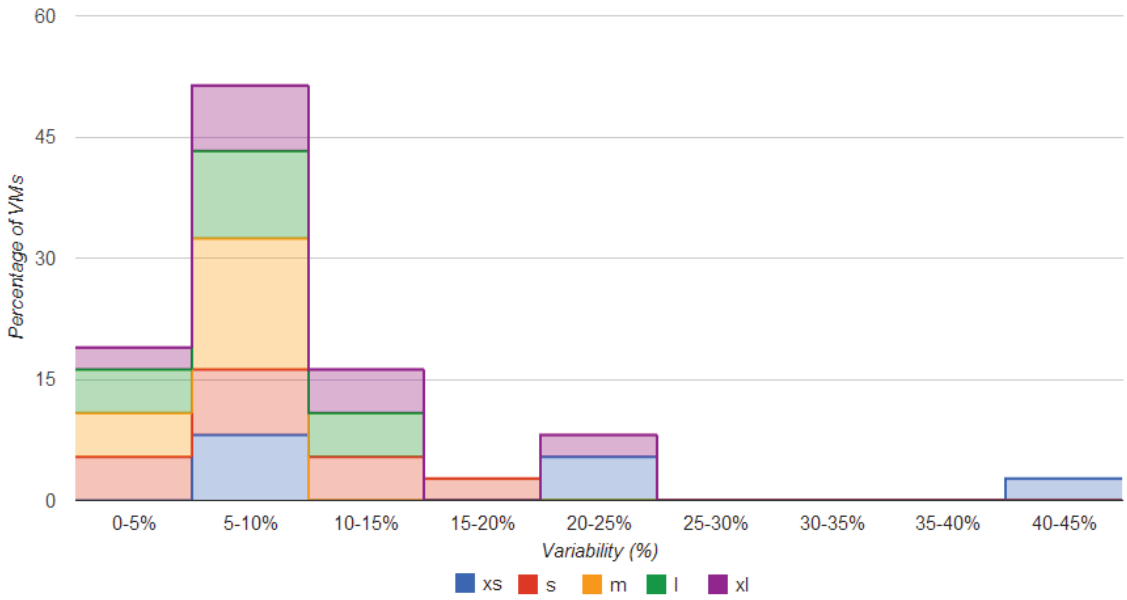

Figure 3.9: Distribution of variability for the measured VMs

selection are presented.

3.3.1 Mean and radar plot as figures of merit

Most of the performance evaluation studies report individual benchmarking results (Table 2.1). In an attempt to express the holistic performance of a Cloud service through a single score, Li *et al.* [LOZC13] propose *Boosting*[1] *Metrics* such as the *mean* (*eg.* arithmetic, geometric, harmonic) and the *radar plot*. The *geometric mean*, by definition, is the nth root of the product of the n units in a data set (Equation 3.1). There is a defect when employing means as boosting metrics: the results from different benchmarks must use the same units. This shortcoming is overcome by using a radar plot.

$$M = \sqrt[n]{\prod_{i=1}^{n} \text{Benchmark}_i} \tag{3.1}$$

A radar plot is a simple graphical tool that can depict three or more quantitative values relative to a central point (Figure 3.10). When benchmark results are expressed in different metrics, Li *et al.* propose two standardization methods to express results over a predefined baseline: Higher is Better (HB) (Equation 3.2) and Lower is Better (LB)

[1]The boosting concept comes from the machine learning field. In Cloud service evaluation, boosting refers to the creation of a measure based on primary metrics that measure individual Cloud service features.

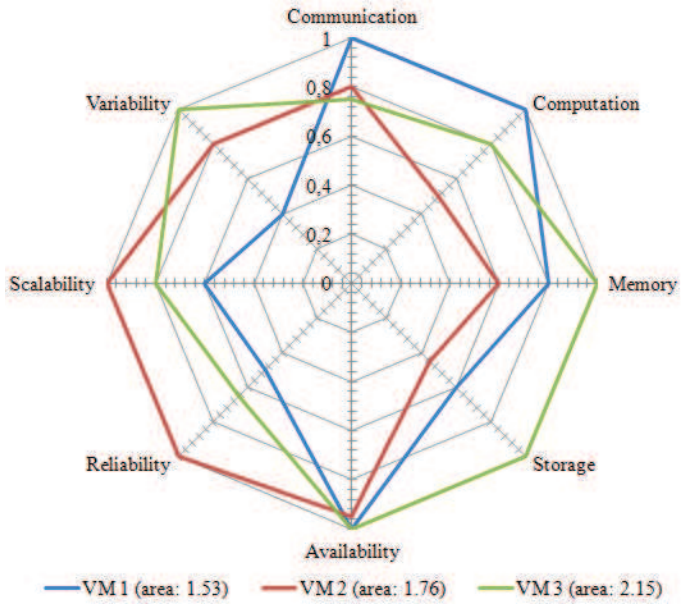

Figure 3.10: Radar plot as a figure of merit

(Equation 3.3). HB (LB) means the higher (the lower) the benchmark result, the better.

$$\text{HB Standardized}_i = \frac{\text{Benchmark}_i}{\text{MAX}(\text{Benchmark}_{1,\ldots,n})} \tag{3.2}$$

$$\text{LB Standardized}_i = \frac{\frac{1}{\text{Benchmark}_i}}{\text{MAX}(\frac{1}{\text{Benchmark}_{1,\ldots,n}})} \tag{3.3}$$

Here HB Standardized$_i$ and LB Standardized$_i$ refer to the standardized ith benchmark result. Thus, the area of the polygon representing n standardized benchmarking results can be considered as a figure of merit of Cloud performance (Equation B.3) [LOZC13].

$$\text{Figure of merit}_{(\text{radar plot})} = \sum_{i=1}^{n} \frac{\sin(\frac{2\pi}{n}) \times \text{Standardized}_i \times \text{Standardized}_{mod(i+1,n)}}{2} \tag{3.4}$$

Although these metrics result in a figure of merit, they present main concerns such as lack of weighting and categorical scores[2].

[2]Score with a limited and usually fixed, number of possible values.

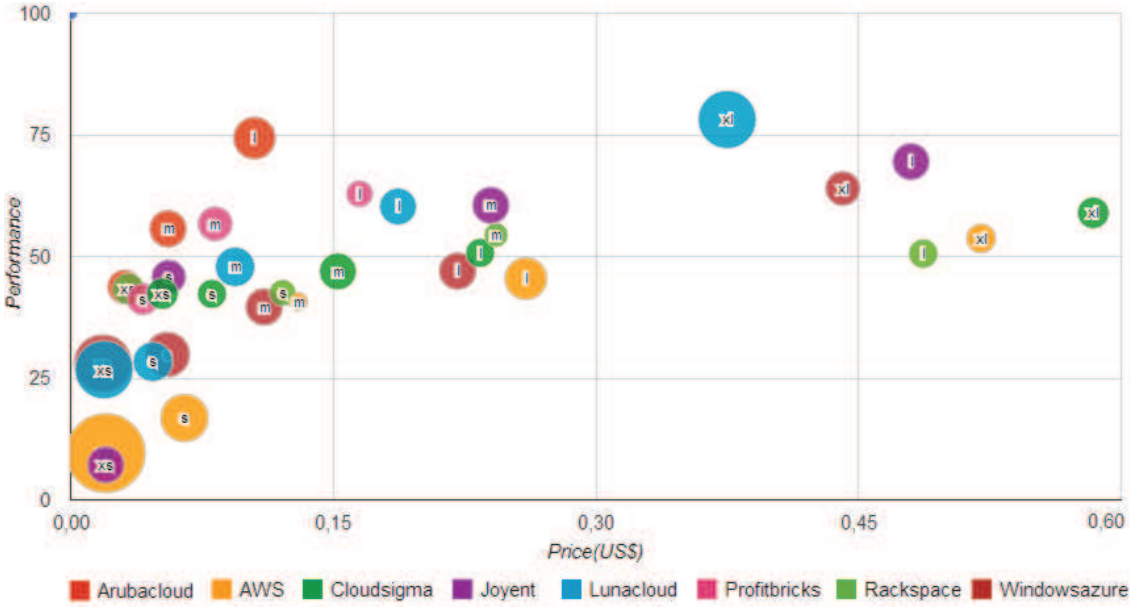

Figure 3.11: Correlation between performance and price for different VM sizes. The variability is represented by the size of the spot. A=1 and B=100.

3.3.2 Simple figure of merit

The simple figure of merit of Cloud performance is a method similar to the one employed by companies reporting Cloud performance such as CloudSpectator or Cloudharmony. It is called simple since it does not take into account the trade-off among the different criteria. In this method, each benchmark result is scaled between two fixed values, A and B. Where A is the lower bound corresponding to the worst performance result (wpr) and B is the upper bound corresponding to the best performance result (bpr). The intermediate values (x_i) are calculated with Equation 3.5. Then, all the scaled values are averaged and a single figure of merit is obtained for each VM configuration. This method has been applied to the data previously reported (*c.f.* Section 3.2) to obtain a figure of merit of Cloud performance (Figure 3.11). More results based on the simple figure of merit method can be found in Appendix A.4.

$$\text{Performance score} = \begin{cases} A + \frac{B-A}{\text{bpr}-\text{wpr}}(x_i - \text{wpr}) & \text{if HB benchmark} \\ B - \frac{B-A}{\text{bpr}-\text{wpr}}(x_i - \text{bpr}) & \text{if LB benchmark} \end{cases} \tag{3.5}$$

3.3.3 Figure of merit based on Analytic Hierarchy Process

Analytic Hierarchy Process (AHP) is a structured technique for analyzing, organizing and solving problems related to Multiple Criteria Decision Making (MCDM) [Saa80]. In AHP, complex problems are simplified and structured by arranging the decision factors

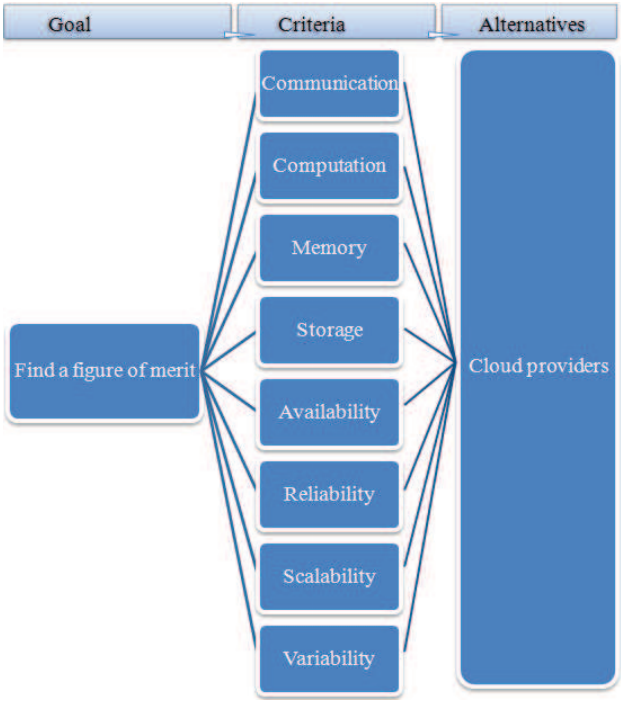

Figure 3.12: Hierarchy of the problem

in a hierarchical structure. The trade-offs among criteria are determined by a pairwise comparison. Unlike the traditional weighted sum-based methods, AHP is based on both subjective and objective evaluation measures. In Cloud computing, AHP has been used to rank Cloud services [GVB13, SZXW13]. In this section, AHP is used to determine the relative merit of members of a set of alternatives. This process consists of three phases: hierarchy structure modeling, judgement of priorities and hierarchical synthesis.

Phase 1: hierarchical structure modeling

In this phase, the problem is defined and the goal is determined. Also, all the criteria that have an influence in resolving the issue are identified, as well as the alternatives that offer an answer to the problem. Both criteria and alternatives are organized in a hierarchical structure. The hierarchy structure used here (Figure 3.12) is based on the performance criteria described previously (*c.f.* Section 2.2.3). The alternatives correspond to the different Cloud providers supported by a Cloud broker. Each alternative represents a set of benchmark results that contains a figure of merit for each criterium. In this case, the goal is to find a figure of merit for a Cloud infrastructure based on performance.

Intensity of importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgement slightly favor one activity over another
5	Essential or strong importance	Experience and judgement strongly favor one activity over another
7	Demonstrated importance	An activity is strongly favored and its dominance demonstrated in practice
9	Absolute importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate values between the two adjacent judgements	When compromise is needed
Reciprocals of above nonzero	If activity i has one of the above nonzero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i	

Table 3.3: Relative rating scale [Saa80]

Phase 2: judgement of priorities

Pairwise comparisons are used to determine the relative importance of each alternative and each criteria. Saaty [Saa80] proposes a relative rating scale (Table 3.3) by which the decision-maker expresses his opinion about the relative importance of one criteria over another. This scale allows one to quantify the pairwise comparisons. This phase leads to the construction of P pairwise comparison matrices of size N-*by*-N, where N is the number of alternatives and P is the total number of criteria. One additional matrix C, the pairwise comparison criteria matrix, is constructed to express the relative weights between each one of the criteria to be evaluated.

Phase 3: hierarchical synthesis

Once all comparisons have been made in phase 2, the numerical probability of each alternative is calculated. This probability determines the likelihood that the alternative has to fulfill the expected goal. This process is applied also to the matrix C that expresses the relative weights between each one of the criteria. The hierarchical synthesis phase is applied to the pairwise comparison matrices as follows:

1. Synthesize the pairwise comparison matrix. Given the pairwise matrix S of size N-*by*-N, the synthesized pairwise comparison matrix (A) is obtained by dividing

N	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Table 3.4: Random Consistency Index (RI) [Saa05]

each value of A by the total of its column, as follows:

$$\forall n \in [1...N], \forall i \in [1...N] : a_{ij} = \frac{s_{ij}}{\sum_{i=1}^{N} s_{in}} \tag{3.6}$$

where a_{ij} is an element of matrix A in row i and column j.

2. Calculate the priority vector (V). The priority vector corresponds to the eigenvector of matrix A. The priority vector can be approximated to the average value of each row of matrix A (Equation 3.7), in order to avoid the mathematical effort required to calculate an eigenvector [Kos91].

$$\forall v \in [1...N] : v_i = \frac{\sum_{i=1}^{N} a_i}{N} \tag{3.7}$$

3. Calculate the maximum eigenvalue (λ_{max}). λ_{max} is calculated by adding the product of each element of vector V by the sum of its corresponding column of matrix S (Equation 3.8).

$$\lambda_{max} = \sum_{j=1}^{N} v_j \cdot \sum_{i=1}^{N} s_{ij} \tag{3.8}$$

4. Calculate the Consistency Index (CI):

$$\text{CI} = \frac{\lambda_{max} - N}{N - 1} \tag{3.9}$$

5. Check the consistency of the pairwise comparison matrix (S). Saaty [Saa05] suggests the Consistency Ratio (CR) in order to determine if the pairwise comparisons made by the decision maker are consistent. For example, consider three criteria x, y and z. If the decision maker has considered $x>y$ and $y>z$, then it would be inconsistent to consider that $x<z$. The CR is calculated as follows:

$$\text{CR} = \frac{\text{CI}}{\text{RI}} \tag{3.10}$$

where the Random Consistency Index (RI) is a fixed value provided by Saaty [Saa05] (Table 3.4). The decisions are considered as consistent when $CR < 0.1$

Finally, the overall performance for each alternative is calculated. For this, a matrix L of size *P-by-H* is formed. In matrix L, each row corresponds to one of the H priority vectors found for each one of the P criteria. The overall performance (OP) corresponds

to the product of the priority vector (G) of matrix C and the matrix L (Equations 3.11 and 3.12). Each element of vector OP corresponds to the performance of one of the H alternatives.

$$\mathrm{OP} = GL \tag{3.11}$$

$$\mathrm{OP} = \begin{pmatrix} g_1 & g_2 & \cdots & g_P \end{pmatrix} \begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,H} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,H} \\ \vdots & \vdots & \ddots & \vdots \\ l_{P,1} & l_{P,2} & \cdots & l_{P,H} \end{pmatrix} \tag{3.12}$$

3.4 Case study: CPU-intensive application

The objective of this case study is to find a single figure of merit of Cloud performance for a CPU-intensive application (*e.g.* file encryption, encoding, scientific computing). In order to compare the different approaches presented previously, a figure of merit with the radar plot has been computed (*c.f.* Section 3.3), the simple and the AHP techniques. The criteria considered were: computation, memory, storage, availability, scalability and variability. In order to calculate a figure of merit based on real data, the performance values obtained previously have been used and the availability values have been obtained from Cloud providers' websites. The availability used is the one for which the Cloud provider will not reimburse the end-user in case of service unavailability. We limited our study to *s*-size VMs. The implementation details per technique are the following:

1. *Radar plot*: benchmark results are standardized with Equations 3.2 and 3.3. A single figure of merit is calculated with Equation B.3 (Figure 3.13a).

2. *Simple figure of merit*: benchmark results are scored with Equation 3.5. Then, the single figure of merit corresponds to the mean value of the scored values of performance (Figure 3.13b).

3. *Figure of merit based on AHP*: the benchmark results are standardized with Equations 3.2 and 3.3. We have found mean values of benchmarks evaluating Cloud performance for the same criteria (*e.g.* in the case of the computation criteria, we calculate the mean value of standardized results of 7zip and C-ray). The pairwise comparison criteria matrix (Table 3.5) considers computation and variability criteria with an equal importance. The computation criterium is also absolutely more important than the storage and scalability criteria. The memory is slightly more important than the storage, availability and scalability. The availability is considered more important than the storage and the scalability. The priority vector represents

Criteria	Computation	Memory	Storage	Availability	Scalability	Variability	Priority vector
Computation	1	6	9	3	9	1	0.3753
Memory	1/6	1	3	3	3	1/6	0.1250
Storage	1/9	1/3	1	1/3	1	1/6	0.0408
Availability	1/3	1/3	3	1	3	1/3	0.1053
Scalability	1/9	1/3	1	1/3	1	1/3	0.0501
Variability	1	6	6	3	3	1	0.3036

Table 3.5: Pairwise comparison criteria. CR = 0.0702 and RI = 1.24.

Rank	Criteria/ Provider	Computation	Memory	Storage	Availability	Scalability	Variability	Overall performance
1	**ARU**	0.2507	0.2304	0.3016	0.1788	0.2726	0.2564	0.2455
6	**AWS**	0.0978	0.1152	0.0912	0.1128	0.1151	0.0995	0.1027
2	**CLO**	0.1319	0.1230	0.1334	0.1233	0.1649	0.1501	0.1371
3	**JOY**	0.1319	0.1334	0.1334	0.1233	0.1457	0.1323	0.1320
7	**LUN**	0.0765	0.0720	0.1061	0.1128	0.0641	0.0853	0.0830
4	**PRO**	0.1284	0.1310	0.1042	0.1128	0.0832	0.1040	0.1164
5	**RAC**	0.1074	0.1100	0.0882	0.1233	0.0890	0.1294	0.1144
8	**WIN**	0.0753	0.0850	0.0419	0.1128	0.0653	0.0431	0.0688

Table 3.6: Overall Cloud performance matrix

the relative importance of each criteria in the single figure of merit.

The priority vectors are calculated for each criterium based on the these mean standardized values. The priority matrix for assessment of the overall Cloud performance (Table 3.6) presents the overall performance for each Cloud provider (Figure 3.13c).

Besides finding a figure of merit of Cloud performance, the cost plays an important role in Cloud service selection. For this, we have considered the performance-price ratio (Figure 3.13d).

The results show that for the three methods to compute a figure of merit, ArubaCloud *s*-size VMs present the best performance among all the evaluated Cloud providers. In the case, of the radar plot and the simple figure of merit approaches the results of performance of Joyent are close to those of ArubaCloud (Figure 3.13a-b). However, with the AHP approach, it can be clearly seen that in the case of a CPU-intensive application, ArubaCloud doubles in performance Joyent (Figure 3.13c).

3.5 Summary

In this chapter, we have evaluated Cloud performance by using micro-benchmarks. The results obtained from benchmarks have been used to calculate a single figure of merit of Cloud performance with the radar plot, the simple and the Analytic Hierarchy Process (AHP) techniques. The advantage of AHP, in comparison with other techniques to calculate figures of merit, is that it is based on pairwise comparisons in which users can

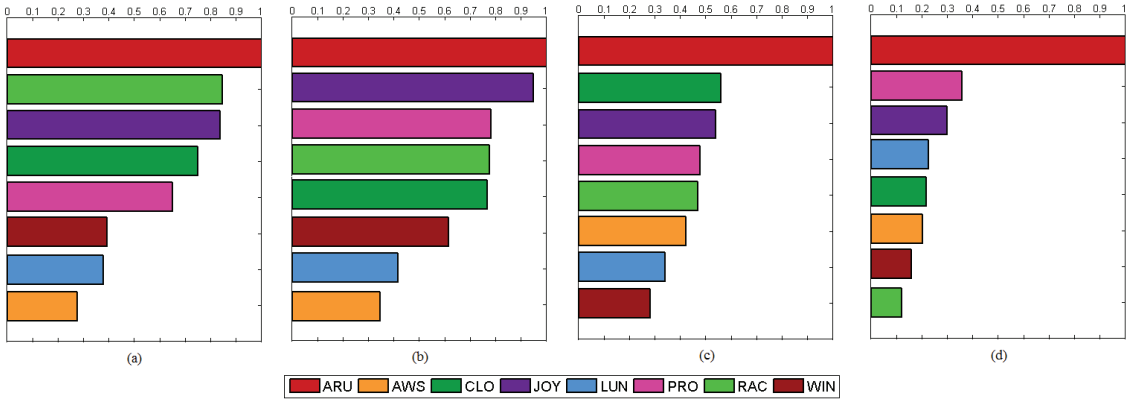

Figure 3.13: Comparison of figures of merit techniques for *s*-VM size. (a) Radar plot (b) Simple figure of merit, A = 0 and B = 1 (c) Figure of merit based on AHP (d) Performance-price ratio based on AHP performance values. The values of the four figures have been normalized in order to ease the comparison.

express the importance of one feature over another. Thus end-users can take into account the requirements an application has in terms of performance criteria.

Chapter 4

An exact approach for optimizing placement in Cloud brokering

Contents

4.1 Introduction

In the near future, Cloud brokers may become the online travel planning companies for Cloud computing. Several years ago, the travel industry was in the same situation as the current Cloud computing industry. Travelers left planning into the hands of travel agencies who had agreements with airline companies and hotels in order to provide an overall value added proposition. This situation changed with the advent of travel planning websites that provide instant online comparisons for millions of flights on over a thousand airlines. Similarly, nowadays consulting companies help end-users with the placement of Cloud infrastructure. In the near future, Cloud brokers may provide not only discovering and comparison of Cloud providers' offers but also automatic and optimal placement of Cloud resources.

Placement in Cloud brokering refers to the techniques to distribute efficiently infrastructure resources across multiple Cloud providers (*c.f.* Section 2.3). Cloud brokers may react to new situations when conditions change, dynamically repositioning or deploying new Cloud infrastructure in order to maintain the performance of end-users' applications. Examples of scenarios in which a Cloud broker may trigger placement algorithms in order to compute a new infrastructure topology include:

- *Changes in Cloud market conditions*: for example, introduction of new VM configurations, change in prices, apparition of a new Cloud provider, implementation of a new pricing model. In this scenario, a Cloud broker could determine the impact of the changes of market conditions on the economies or performance of end-users' applications. In the case of a positive impact, the end-user can be advised to migrate its Cloud infrastructure.
- *Unexpected changes in Cloud infrastructure*: outages may strongly impact economies of end-users' running Cloud applications. Although Cloud providers offer economic compensation to end-users having experienced an outage, in most of the cases, this compensation is negligible in comparison to the impact of having a service unavailable (*e.g.* an e-commerce website down). Thus, Cloud brokers may not only automatically redeploy infrastructure in recovery scenarios but also minimize the time an application is inaccessible.

This chapter is organized as follows. Goal programming, a technique to solve Multiple Criteria Decision Making (MCDM) problems is briefly described in Section 4.2. An exact approach for optimizing placement in Cloud brokering is presented in Section 4.3. A case study considering an online trading platform is presented in Section 4.4.

4.2 Goal programming

The optimization goal in MCDM problems is to find an efficient (but not necessarily an optimum) solution by considering multiple objectives (or goals) that can possibly conflict with each other. Thus, MCDM problems contrast with Linear Programming (LP) problems which optimizes a single linear objective. Here, we consider goal programming as a technique to solve MCDM problems. Goal programming is usually carried out using either the *weighted* or the *preemptive* method.

The weighted method transforms a MCDM problem into a standard LP. A weighted objective function corresponds to a weighted sum of functions representing the multiple objectives of the problem. The weights determine the priority of each objective. Although the computation of the weighted method is easy, the main drawbacks are:

- The weights selection is subjective and may result in under- or over-rating the

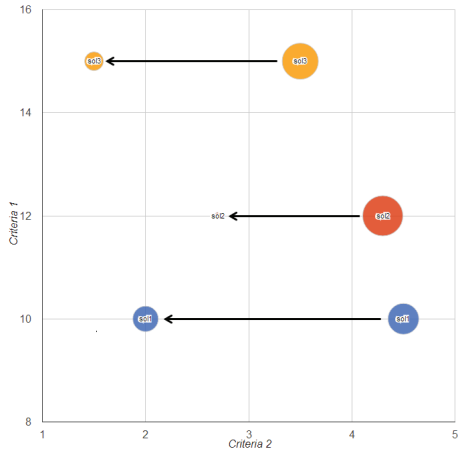

Figure 4.1: Preemptive method. Optimization priority: Criteria 1, Criteria 2, Criteria 3 (represented by the size of the spot).

contribution of objectives.

- The objectives may be expressed in different metrics or different order of magnitudes that prevent a straightforward calculation of the objective function.

The preemptive method considers a MCDM problem as a set of multiple LPs with different priorities assigned by the end-user. Thus, each LP is optimized one at a time from the highest to the lowest priority (Figure 4.1). Between LP executions, the optimum value is added as a constraint to the successive LP model. This guarantees that the optimum value of a higher priority objective is not degraded by a lower priority objective. This process continues until the lowest priority is optimized. In cases where a limited amount of degradation is acceptable, the constraint is added as an inequality that allows higher priority solutions to be in the near neighborhood of the optimal solution.

4.3 An exact approach for the Placement problem

The Cloud VM placement problem in Cloud brokering can be represented as a constrained Knapsack problem: Given a set of VMs, each with a configuration, a price, a performance, determine the number of each VM configuration to provide so that the provisioning infrastructure is more than or equal to the end-user request (*i.e.* request is satisfied) and the cost of the Cloud infrastructure is as small as possible (in the case of cost optimization). The Cloud placement problem is formulated as an Mixed-Integer Linear Programming (MILP) problem and the preemptive goal programming analysis is performed to solve this problem.

4.3.1 Parameters

- *End-user request parameters:*
  - *reqCPU*: number of vCPUs.
  - *reqMEM*: memory capacity.
  - *reqSTO*: storage capacity.
  - *reqNET*: network capacity.
  - *reqRTT*: average latency between the Cloud providers and the application customers.
  - *reqAVA*: average availability.
  - *reqREL*: average reliability.
  - *reqSCA*: average VM provisioning time.
  - *reqVAR*: average variability.
  - *reqPER*: performance required by the end-user.
  - $LOCmax_k$: maximum percentage of resources that can be allocated to Cloud provider k.
  - *VMmax*: maximum number of VMs.
  - *Pricing model*: on-demand, reserved or spot.
- *Cloud provider parameters:*
  - V: j-by-k matrix composed of v_{jk} elements. Where $v_{jk} = 1$ if and only if the VM configuration j exists at the Cloud provider k. $v_{jk} = 0$ otherwise.
  - CPU_{jk}: the number of vCPUs of the VM configuration v_{jk}.
  - MEM_{jk}: the memory capacity of the VM configuration v_{jk}.
  - STO_{jk}: the storage capacity of the VM configuration v_{jk}.
  - NET_{jk}: the bandwidth capacity of the VM configuration v_{jk}.
  - $Price_{jk}$: the price per unity of time for running a VM configuration of type v_{jk}.
- *Cloud broker parameters:* Parameters measured or calculated by the Cloud broker.
  - p: index of the smallest VM configuration (in terms of computing, memory and storage).

– η_k: the number of VMs of type p that guarantees the fulfillment of the request for the Cloud provider k.

$$\forall k \in [1, K]: \quad \eta_k = max\left(\left\lceil \frac{reqCPU}{CPU_{pk}} \right\rceil, \left\lceil \frac{reqMEM}{MEM_{pk}} \right\rceil, \left\lceil \frac{reqSTO}{STO_{pk}} \right\rceil\right) \tag{4.1}$$

– N: as we consider Cloud providers with unlimited resources, N is an upper bound that limits the set of solutions. Thus, N represents the maximum number of any kind of VM configuration used to fulfill the request. This parameter is set in case of the $VMmax$ is not specified by the end-user.

$$N = max(\eta_k) \tag{4.2}$$

– RTT_k: the latency of the Cloud provider k.

– α_k : average availability of a Cloud provider k.

– β_k : average reliability of a Cloud provider k.

– γ_{jk}: average time to provision a VM of type j at Cloud provider k.

– cv_{jk}: average variability of a VM of type j at Cloud provider k.

– $Performance_{jk}$: the performance of a VM configuration of type v_{jk}.

4.3.2 Variables

- *Binary variables:*

  – $x_{jk}^n = 1$: if and only if the VM n of type j is used and belongs to the Cloud provider k. $x_{jk}^n = 0$ otherwise.

- *Real variables:*

  – TCC: Total Computing Capacity. Amount of computing capacity for a particular solution.

$$TCC = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} CPU_{jk} x_{jk}^n \tag{4.3}$$

  – TMC: Total Memory Capacity. Amount of memory capacity for a particular solution.

$$TMC = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} MEM_{jk} x_{jk}^n \tag{4.4}$$

– *TSC*: Total Storage Capacity. Amount of storage capacity for a particular solution.

$$TSC = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} STO_{jk} x_{jk}^{n} \tag{4.5}$$

– *TVM*: Total of VMs of a particular solution.

$$TVM = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} x_{jk}^{n} \tag{4.6}$$

– *TDC*: Total Deployment Cost. Total cost for deploying an infrastructure across multiple Cloud providers.

$$TDC = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} Price_{jk} x_{jk}^{n} \tag{4.7}$$

– *TP*: Total performance of a particular solution.

$$TP = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} Performance_{jk} x_{jk}^{n} \tag{4.8}$$

– *TDT*: Total Deployment Time. Total time for deploying an infrastructure across multiple Cloud providers.

$$TDT = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_{jk} x_{jk}^{n} \tag{4.9}$$

– *TV*: Total Variability. Total variability for a particular solution.

$$TV = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} cv_{jk} x_{jk}^{n} \tag{4.10}$$

4.3.3 Goal

Preemptive goal programming analysis is performed to solve the MILP depending on the scenario. For instance, in a disaster recovery scenario the priority of the criteria is as follows:

1. Minimize the real variable *TDT*.
2. Minimize the real variable *TDC* constrained by the minimal deployment time previously obtained.

3. Maximize the real variable TP constrained by the minimal deployment time and the total deployment cost previously obtained.

This method guarantees a minimal deployment time and suboptimal cost and performance values of the infrastructure to be provisioned.

4.3.4 Constraints

The constraints associated are the following:

- *Physical constraint*: this constraint guarantees VMs are allocated to an existing VM configuration.

$$x_{jk}^{n} \leq v_{jk} \tag{4.11}$$

- *VM configuration constraints*:

$$TCC \geq reqCPU \tag{4.12}$$

$$TMC \geq reqMEM \tag{4.13}$$

$$TSC \geq reqHD \tag{4.14}$$

$$VMmax \geq TVM \tag{4.15}$$

- *Load Balancing constraints*:

$$\forall k \in [1, K]: \quad \sum_{j=1}^{J} \sum_{n=1}^{N} CPU_{jk} x_{jk}^{n} \leq LOCmax_{k} \cdot TCC \tag{4.16}$$

$$\sum_{j=1}^{J} \sum_{n=1}^{N} MEM_{jk} x_{jk}^{n} \leq LOCmax_{k} \cdot TMC \tag{4.17}$$

$$\sum_{j=1}^{J} \sum_{n=1}^{N} HD_{jk} x_{jk}^{n} \leq LOCmax_{k} \cdot TSC \tag{4.18}$$

- *Availability and reliability constraint*:

$$\begin{gathered}\forall k \in [1,K] : y_k = \sum_{j=1}^{J}\sum_{n=1}^{N} x_{jk}^n \Rightarrow \\ \sum_{k=1}^{K} \alpha_k \cdot y_k \leq reqAVA \cdot \sum_{k=1}^{K} y_k \\ \sum_{k=1}^{K} \beta_k \cdot y_k \leq reqREL \cdot \sum_{k=1}^{K} y_k\end{gathered} \tag{4.19}$$

- *Latency constraint*:

$$\sum_{k=1}^{K} RTT_k \cdot \sum_{j=1}^{J}\sum_{n=1}^{N} x_{jk}^n \leq reqRTT \cdot TVM \tag{4.20}$$

- *Scalability constraint*:

$$\sum_{j=1}^{J}\sum_{k=1}^{K} \gamma_{jk} \cdot \sum_{n=1}^{N} x_{jk}^n \leq reqSCA \cdot TVM \tag{4.21}$$

- *Variability constraint*:

$$\sum_{j=1}^{J}\sum_{k=1}^{K} cv_{jk} \cdot \sum_{n=1}^{N} x_{jk}^n \leq reqVAR \cdot TVM \tag{4.22}$$

4.4 Case study: Online trading platform

Bezimie is a London based company that provides an online trading platform. Using Bezimie's application, traders can purchase and sell stocks and currencies easily through its web interface. The main competitive advantage of Bezimie regarding other online trading platforms is the low latency[1] at the moment of placing market orders or reporting prices quotes of stocks and currencies to traders. Bezimie manages its Cloud infrastructure with the help of the CompatibleOne Cloud broker. The current Cloud infrastructure topology of Bezimie consists of multiple VMs deployed across two providers (Amazon and Rackspace) with datacenters in Ireland and England. The Cloud providers have been chosen due to their proximity to Bezimie's clients, resulting in the low average latency ideal for online trading (Table 4.1).

[1]The network latency is particularly important in financial instruments with a high price variation (volatility). A few cent price variation may represent large amounts of money when trading in high volume. Moreover, higher latency connections are more prone to packet delivery delays and loss.

Cloud provider	RTT to England (ms)	RTT to France (ms)
ARU	127	82
AWS	95	112
CLO	135	105
JOY	105	101
LUN	110	91
PRO	130	110
RAC	85	97
WIN	122	123

Table 4.1: RTT from Cloud providers to current and future Bezimie's client portfolio

Bezimie is planning to expand its portfolio of clients to France. After some tests, Bezimie's IT department notes that the latency of French traders to its UK-based Cloud infrastructure is acceptable ($\leq$ 125 ms) for online trading but greater than the latency of Zimie ($\leq$ 110 ms) its French counterpart and direct competitor. Therefore, in order to be competitive in the French market, Bezimie's IT department compares different solutions to serve its French traders with help of the CompatibleOne broker. The first optimization priority is to minimize the latency between Cloud providers and traders; the second priority is to minimize the cost of the requested infrastructure; the third is to maximize the performance of the future acquired VMs. Bezimie's IT department chooses the best performing solution (19.6) with the lowest latency in France ($\leq$ 82 ms) at the lowest cost (2.5 US$ per hour) for serving its French traders (Figure 4.2). However, this solution places all the infrastructure to serve French traders into one Cloud provider (ArubaCloud). This represents a serious defect in the case of Cloud service outages.

Bezimie's IT department simulates also disaster recovery scenarios through the CompatibleOne Cloud broker. In disaster scenarios, the main priority for Bezimie is to minimize the time the service is offline while keeping an acceptable latency. The second priority is to minimize the cost of the required infrastructure. The solutions for provisioning time optimization in case of an ArubaCloud outage are presented in Figure 4.3. The figure shows the two optimization stages for different LOCmax values. On the right, the solutions resulting from the first optimization stage (minimization of the provisioning time). On the left, the solutions resulting from the cost optimization stage. Note that cost optimization brings cheaper but higher latency and more variable solutions.

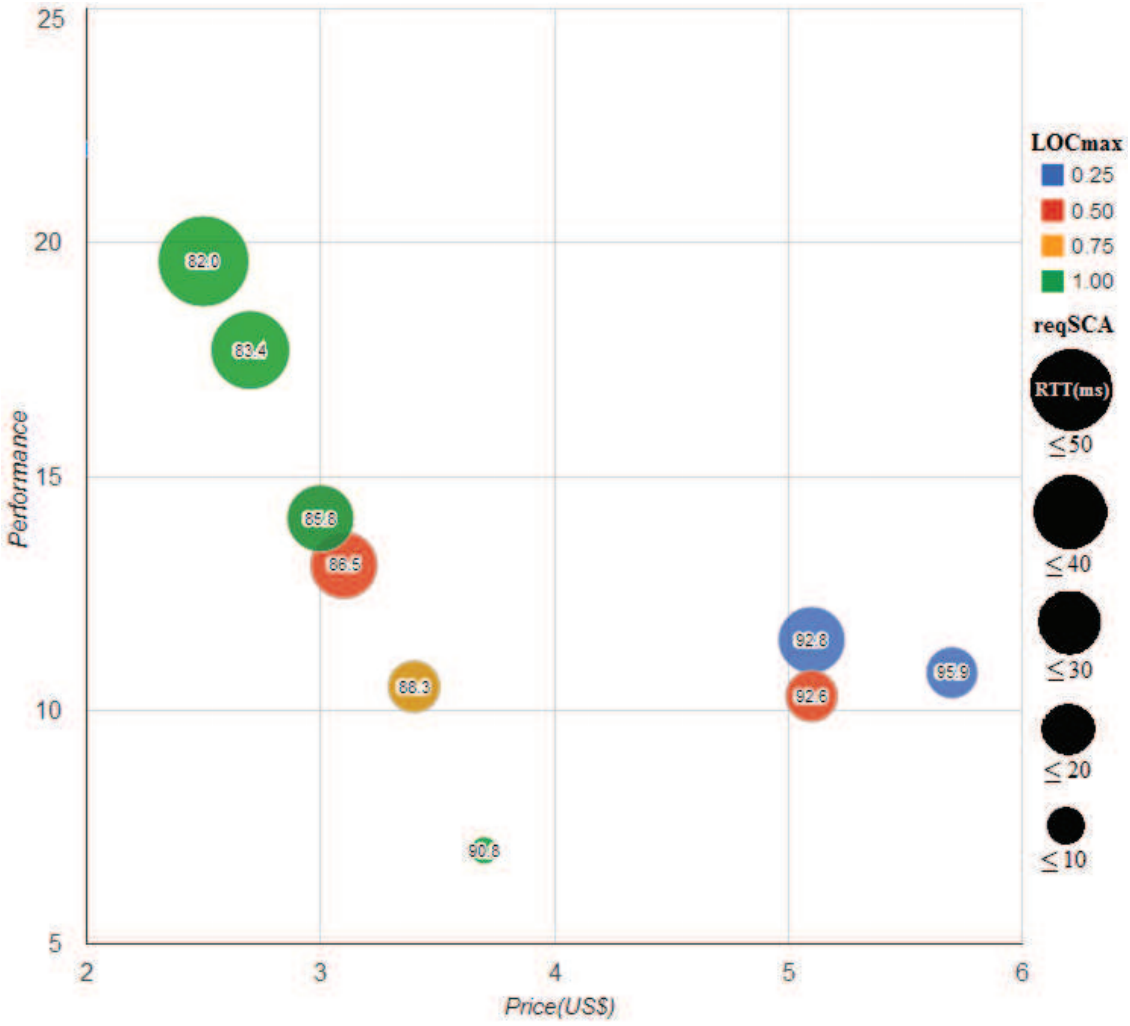

Figure 4.2: Solutions for latency preemptive optimization. The size of the spot represents the average provisioning time in seconds (reqSCA). The figure of merit here used is the same obtained in the previous case study (*c.f.* section 3.4). Parameters: reqCPU = 80, reqMEM = 60, reqSTO = 300, reqRTT $\leq$ 110ms.

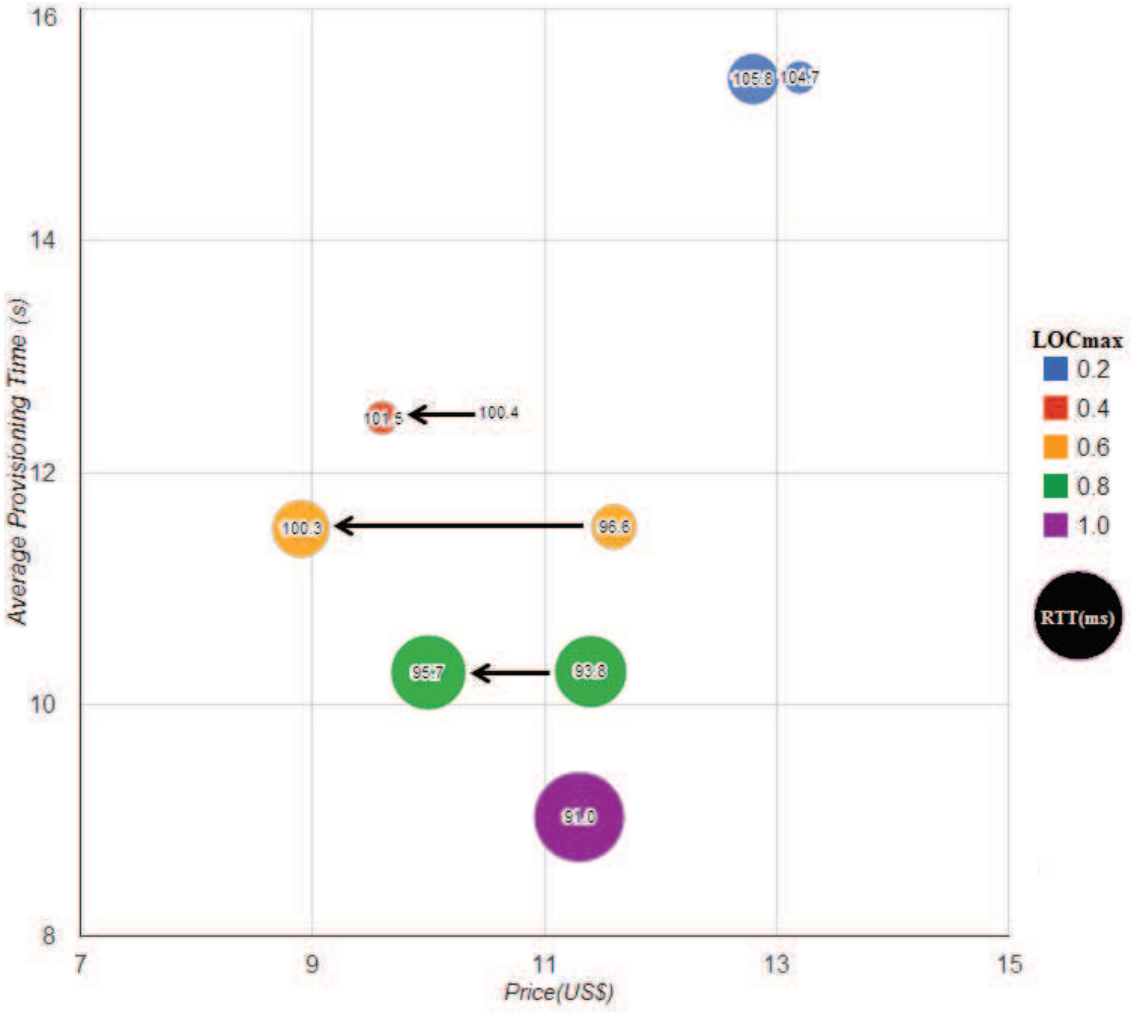

Figure 4.3: Solutions for provisioning time preemptive optimization. The size of the spots represents the solution variability. Parameters: reqPER = 20, reqRTT $\leq$ 110ms.

Part II

A new pricing model in Cloud brokering

Chapter 5

The Pay-as-you-book pricing model

Contents

5.1 Introduction

The most popular pricing models used by the current Cloud providers are: *Pay-as-you-go* and *subscription-based*. *Pay-as-you-go* involves a high price per unity hour but does not require long-term commitment. On the contrary, subscription-based pricing models result cheaper than pay-as-you-go in the long-term but normally require a long-term commitment and associated vendor lock-in. In this chapter, the current employed pricing

models in Cloud computing are briefly described (Section 5.2). The Advance Reservations (ARs), an efficient way to guarantee the availability of a given amount of resources for use at a specific time in the future is studied (Section 5.3). Then, the concept of *pay-as-you-book* (Section 5.4), a novel manner of acquiring Cloud resources in advance for future use based on ARs is presented. Pay-as-you-book combines the main advantages presented in pay-as-you-go and subscription-based pricing models: no long-term commitment and low cost, respectively. At the end of this chapter, a case study comparing the impact of different resource allocation policies on the economies of a Virtual Cloud Provider (VCP) is developed (Section 5.5).

5.2 Pricing models in Cloud computing

Several economic models from other fields of study have been proposed for Grid Computing [BAGS02a]. The commodity market, posted price, tender, bargaining and auction models are among the commonly studied economic models employed for managing the resources in the Cloud [BAGS02b]. However, most of them have not been implemented by current Cloud providers. *Pay-as-you-go* and *subscription-based* pricing models are among the most popular cloud pricing models applied by current Cloud providers [WABS09]. In the pay-as-you-go model users pay a value proportional to their resource consumption, while in subscription-based pricing models users must commit to use the service for a given period of time, in exchange of paying a lower price per hour than in pay-as-you-go. Generally, purchased resources through subscription-based pricing models have priority in terms of availability over resources acquired through pay-as-you-go. The following Cloud pricing models are currently deployed by IaaS Cloud providers:

1. *Freemium:* a product or a service is free of charge, but users must to pay for advanced features. The product or service may be restricted by time, capacity, customer class, features, and so on (*e.g.* Amazon EC2 *Free Tier Instances*).

2. *Usage duration or pay-as-you-go:* users pay a value proportional to their resource consumption (*e.g.* Amazon EC2 *On-Demand Instances*).

3. *Subscription-based:* users must commit to use the service for a given period of time, in exchange they pay a lower price in the long term than in pay-as-you-go. This pricing model allows Cloud providers to foresee the utilization of their Cloud infrastructure in advance and to speed up their Return On Investment (ROI). The resource allocation, in this pricing model, is based on ARs; Cloud providers lock resources and guarantee their future availability to end-users [LRY+11]. Subscription-based pricing model may be divided into three categories:

    - Flat-fee or flat-rate: users are charged a fixed fee for a given period of time, regardless of the resource utilization (*e.g.* Amazon EC2 *Heavy Utilization*

Reserved Instances).

- Subscription with quota: users are charged a fixed fee to subscribe the service and are given an usage quota. If the quota is exceeded, there is an additional charge.
- Subscription without quota: users are charged a fixed fee to subscribe the service plus an additional extra charge depending on usage (*e.g.* Amazon EC2 *Light* and *Medium Utilization Reserved Instances*).

4. *Market-based:* users bid for computing power, resources are allocated if the bid exceeds the price fixed by the Cloud provider (*e.g.* Amazon EC2 *Spot Instances*). This pricing model is used by Cloud providers to sell spare Cloud computing capacity.

Users select a pricing model based on their needs (such as computation power, memory and storage capacities, QoS, execution time, budget and so on). Thus, users with time-constrained tasks would be more interested in purchasing *flat-rate* VMs, in order to assure computing power at anytime. On the contrary, users without time-constrained tasks would be willing to acquire VMs through the Market-based pricing model. In case of fluctuating and unpredictable loads, VMs are purchased through the pay-as-you-go model.

5.3 Advance Reservations

Advance Reservations (ARs) have been introduced as an efficient way to guarantee the availability of a given amount of resources for use at a specific time in the future. Hotel room bookings are a very well known example of ARs. In hotel room bookings, an AR is described by at least three parameters: numbers of rooms to be booked, and the check-in and check-out dates. AR mechanisms have been applied to several problems of resource sharing in computer science such as bandwidth reservation, job scheduling and VM scheduling. In the following, a classification of some studies dealing with ARs applied to computer science is presented.

5.3.1 Advance Reservation specified by Cloud providers

This type of AR is tightly related to the subscription-based pricing model, widely proposed by Cloud providers (Section 5.2). This type of reservation operates on a time-interval basis. At the beginning of each time-interval, the end-user may adjust the amount of resources to be reserved by the Cloud provider for the next time-interval. Published research studies can be classified into short-term reservation plans [NFL12b, NFL12a] (*e.g.* fine granularity of 10-minute/1-hour time-intervals) and long-term reservation plans (*e.g.* multi-year time-intervals) [SAMVML11, CLN12].

Niu, D. *et al.* [NFL12b] investigated pricing policies for guaranteed bandwidth reservation in the Cloud on a short-term basis such as hours or tens of minutes. Requests are characterized by an estimated average bandwidth requirement, its variability, and the percentage of the traffic flow to be satisfied with guaranteed bandwidth. As for the Cloud provider, it computes the current bandwidth reservation in order to guarantee the required performance in a probabilistic way. It also decides on the reservation fee taking into account the burstiness and the time correlation of the various requests. A similar problem where a broker is introduced between the Cloud providers and the end-users is also investigated by Niu, D. *et al.* [NFL12a]. While the broker sells guarantees to end-users individually, it jointly reserves bandwidth from multiple Cloud providers for the mixed demand, exploiting statistical multiplexing to save reservation cost. The problem has been solved using a game theory approach where the equilibrium bandwidth price depends on the demand expectation, its burstiness as well as its correlation to the market.

The long-term reservation plan was first studied by San-Aniceto, I. *et al.* [SAMVML11]. This approach considered a single Cloud provider and proposed an algorithm that selects the number of VMs to be reserved by an end-user while deploying a service in the Cloud. In order to cope with request fluctuations and unpredictability, additional resources may be dynamically provisioned with an on-demand plan. The proposed algorithm minimizes the global cost of using a mixture of reserved and on-demand VMs by taking advantage of the different pricing models within the same provider. Chaisiri, S. *et al.* [CLN12] generalized the problem to the context of multiple Cloud providers taking into account the uncertainty on end-users future requests and providers' resource prices. They formulated the problem as an integer stochastic program and solved it numerically using various approaches.

5.3.2 Advance Reservation specified by end-users

In this type of AR, end-users have a higher flexibility as they can specify, in addition to their capacity requirements, various time constraints associated with the execution of their tasks. Time constraints can be expressed in terms of various parameters such as start-time, completion time, duration and task deadline. Thus end-users have the opportunity to reserve in advance the estimated required resources for the completion of their tasks without any further commitment. The AR window is defined as the time-interval delimited by the start-time and the deadline of a given AR request. ARs specified by end-users can be classified into the following three categories.

Strict start and completion time

This type of AR is characterized by a duration equal to its AR window. In other words, end-users require the resources at a specified exact time in the future and for a specified duration (Figure 5.1). This type of AR does not leave any flexibility to the Cloud

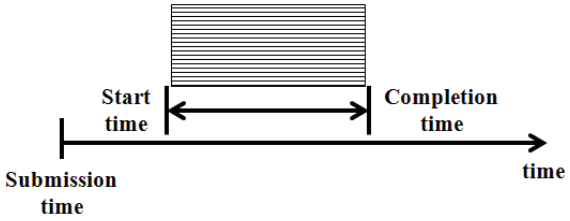

Figure 5.1: Strict start and completion time Advance Reservation

provider to reschedule the AR. Several studies have shown that ARs with strict start and completion times lead to high fragmentation of the resource availability by increasing the number of time intervals that are left unused [SFT00, THW02]. In the case of Cloud computing VMs, these time intervals can be used by other types of requests such as spot or on-demand VMs.

Aoun, R. *et al.* [ADG10] investigated the provisioning of computing, storage, and networking resources in order to satisfy AR requests. They considered several basic services and highlighted how distributed data storage and multicast data transfer can satisfy a larger number of end-users and improve resource utilization of Cloud providers. In further studies, the business model of the aforementioned problem has been investigated [AG09c]. The authors proposed and compared three pricing strategies assessing the expectations of both end-users and Cloud providers.

Flexible start but strict completion time

This type of AR is characterized by a higher flexibility than the former as the AR window is larger than its execution time. However, these ARs are time-critical and, if accepted, the Cloud provider must ensure that they will complete prior to their firm deadline (Figure 5.2). Thus, Cloud providers may use various mechanisms to efficiently arrange, manage, and monitor their resources. For instance, Lu, K. *et al.* [LRY+11] introduced a model based on computational geometry that allows Cloud providers to record and efficiently verify the availability of their resources during the SLA negotiation and planning phase. According to this model, when the Cloud provider lacks resources, a flexible alternative solution, referred to as counter-offer, can be generated in order to satisfy the end-user. Hence, the Cloud provider's reputation can be enhanced by improving its ability to satisfy as many end-users as possible leading to higher resource utilization and consequently higher profits. Venugopal, S. *et al.* [VCB08] investigated a negotiation mechanism that allows both parties (Cloud providers and end-users) to modify the SLA or to make counter proposals in order to converge to a mutually acceptable agreement. In the investigated scenarios, once the SLA has been agreed upon, the Cloud provider has to execute the task at the specified time. Numerical simulations have been carried out to highlight the benefit brought by time-flexible AR requests. Kaushik, N. *et al.* [KFC06] investigated

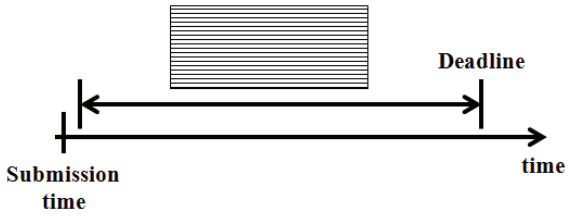

Figure 5.2: Flexible start but strict completion time Advance Reservation

the impact of the AR window size on the blocking probability and the resource utilization for various models of inter-arrival and service times under the first-come-first-served scheduling policy.

Aoun, R. *et al.* [AG09a] investigated the resource provisioning problem in a market-oriented Cloud considering ARs with flexible windows, the size of the AR window being a function of the requirements and the budgets of end-users. The aim of this study is to propose a fair management algorithm that guarantees the QoS requirements of end-users while increasing the expected benefit of Cloud providers. For this purpose, the authors introduced a weighted cost function that enables service differentiation relying on disparity in time constraints of the requests. An exact linear formulation [AG09a] as well as a heuristic approach [AG09b] have been considered for the numerical performance evaluation. Instead of charging fixed prices, Yeo, C.S. *et al.* [YVCB10] propose to automatically adjust the price for accessing the resources, whenever necessary, in order to increase the Cloud provider revenues. By charging variable prices, Cloud providers can give incentives to end-users with less urgent requirements to shift their use to off-peak periods to benefit from lower prices. As the prices are adjusted based on the expected workload and the resource availability, ARs submitted a long time in advance are privileged with cheaper prices compared to late ARs.

Similar investigations have been carried out in a slightly different environment. The new environment allows the Cloud provider to modify the execution schedule of already accepted ARs in order to accommodate new requests right up until each execution starts [NBB07]. Such rescheduling of existing ARs is carried out while respecting the deadline constraints specified in the SLA. The authors have shown that this mechanism can mitigate the negative effects of ARs and improve the performance of reservation-based schedulers as it tends to reduce the amount of time intervals where resources remain free. Another solution to improve resource utilization is to make use of comprehensive overbooking which is particularly efficient in scenarios with no-show policy, AR cancelation [SKB08], and over-estimated execution time of ARs [BB11]. In this context, rescheduling existing ARs may allow overbooked ARs to get access to the resources during their full execution period if previous ARs do not show or finish earlier. The Earliest Deadline First scheduler has been shown to provide probabilistic real-time guarantees for ARs over time-shared machines [KKV+09]. With this scheduling strategy, an admission

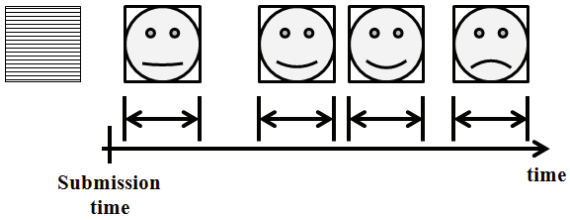

Figure 5.3: Flexible start and completion time Advance Reservation. The mood represents the level of end-user satisfaction for a specific interval.

control policy is developed where new AR requests are accepted if they do not break the QoS constraints of previously accepted reservations. This can be achieved for instance by changing the priority of the running ARs to ensure that the execution completes prior to its deadline.

Flexible start and completion time

This type of AR is also characterized by a high flexibility. However, the AR window is not clearly defined. Instead of defining a start-time and a firm deadline for the execution of each AR, the end-user provides a set of time-intervals along with its preferences represented by a utility function (Figure 5.3). The utility function represents the level of satisfaction that the end-user will experience as a result of the negotiation outcome. This satisfaction may depend on several parameters such as the time of execution, the price of the resources, the delays and the QoS requirements. Not being able to reach an agreement is the worst possible outcome as the end-user receives a null utility from the rejected request. Dynamic pricing based on resource utilization and end-users classification was introduced by Püschel, T. *et al.* [PN09]. Such dynamic pricing strategies allow adapting the price to set incentives for using the resources during off-peak periods. Two different approaches, which are already well established in other areas, are compared by Meinl, T. *et al.* [MAT10] namely, reservation realized by derivative markets in a perfect competition Cloud providers environment and by yield management techniques assuming an imperfect competition environment. The authors analyze the different requirements in order to apply the proposed approaches in the Cloud and provide models to derive the suitable reservation price. Son, S. *et al.* [SS12] introduced a bilateral negotiation mechanism for Cloud service reservation that simultaneously considers price and execution time. Numerical simulations have been used to compare the proposed mechanism to traditional pricing models used by current Cloud providers, namely fixed-prices for on-demand and reserved VMs, and variable prices for spot VMs. The Time-of-Use pricing policy has been investigated by Saure, D. *et al.* [SSQ+10]. According to this policy, the price of accessing resources is totally independent from the utilization ratio of the requested resources but varies within a day. The optimal pricing strategy that maximizes the end-user satisfaction is derived.

In under-estimated ARs, ARs will run for a longer period than expected. Yeo, C.S. *et al.* [YVCB10] deal with the problem of under-estimated ARs with *flexible start but strict completion time*. However, in order to enforce future scheduled ARs, ARs are killed once the time period of reservation expires. In our approach, decisions on whether to kill or keep ARs are made by evaluating ARs' SLA constraints.

5.4 Pay-as-you-book

Pay-as-you-book is a pricing model between pay-as-you-go and subscription-based. Pay-as-you-book consists of paying and reserving time-slots of VMs in advance without a fixed fee to subscribe to the service and without a long-term commitment, avoiding vendor lock-in, while obtaining lower prices than in pay-as-you-go. Thus, combining the advantages of pay-as-you-go and subscription-based pricing models. Another advantage of pay-as-you-book is a fixed user cost provision, due to the fact that users pay what they have reserved. This also represents an advantage for Cloud providers, that could substantively reduce or avoid the use of predictive analytic techniques (*e.g.* modeling, game theory, machine learning, and data mining) to determine utilization patterns. Table 5.1 presents a comparison between the current popular pricing models of Cloud providers and pay-as-you-book.

Pay-as-you-book may be applied in scenarios with predictable workloads [HY10] such as:

- **Time-of-day patterns:** Scenarios with recurring cycles in users' resources consumption based on people's behavior, *e.g.* the consumption of IT resources by users of a company can be easily predicted and described by R resources between 8AM to 5PM from Monday to Friday.
- **Industry-specific variability:** Scenarios with predictable variability based on recurrent events, such as tax season, FIFA World Cup and gift purchases for Christmas.

In pay-as-you-book, an AR Ω^i can be modeled by a set of VMs ω_j^i. Each VM ω_j^i is meant to be used during a specific period of time (*"strict start and completion times"*) and is represented by the tuple $(\alpha_j^i, \beta_j^i, \gamma_j^i)$, where α_j^i denotes the start-time of the VM, β_j^i its stop-time estimated by the end-user, and γ_j^i its real stop-time. An AR is accepted if the set of VMs described in it can be provisioned.

5.4.1 Initial scheduling of Advance Reservations

Since ARs are made prior to VM utilization, the Cloud provider can use various scheduling approaches in order to optimize the resource utilization of its infrastructure and conse-

Feature	Pay-as-you-go	Subscription-based	Pay-as-you-book
Cost	High	Low	Medium
User Cost Provision	Variable	Variable/fixed	Fixed if no under-estimated ARs
Reimbursement in case of service unavailability	None	Percentage of the user fees	X times reservation value
Payment terms	In-arrears	Up-front	Up-front or in-arrears
Term commitments	None	Long (From months to years)	Short (Duration of the reservation)
Availability during periods of very high demand	Low	High	Depends on Cloud provider's policies
Use of predictive analytics	Unpredictable usage patterns	Necessary and done by Cloud provider	Not necessary, prediction done by the end-user
Type of applications	Unpredictable workloads, spiky	Predictable and continuous usage	Very predictable usage

Table 5.1: Most used pricing models compared with pay-as-you-book

quently increases its revenues. At this stage, the Cloud provider has knowledge only of the execution time estimated by end-users. Even though these estimations may be imprecise, the Cloud provider has to decide whether to accept ($\varpi^i = 1$) or reject ($\varpi^i = 0$) each AR Ω^i depending on its resource availability.

The initial scheduling problem can be formulated as follows. Given the number $\mathcal{N}$ of available VMs and the set of $\mathcal{M}$ ARs, the Cloud provider has to determine, for each accepted AR, the physical machine that will host it. This should be carried out while respecting the limited resources of the Cloud provider and the fixed start and completion times estimated by the end-users. The main objective of the Cloud provider is to maximize the utilization of its resources which can be expressed mathematically as:

$$\mathcal{G} = \sum_{i=1}^{\mathcal{M}} \varpi i \times \sum_{\forall j} \left(\beta_j^i - \alpha_j^i \right) \tag{5.1}$$

The choice of the type of the initial scheduling algorithm and its setup depends on the provider goals. In the case of a Cloud provider, the resource allocation goal may aim, for instance, to minimize the number of physical machines used to host the VMs in order to reduce the power consumption, thus reducing the operational expenditures. In the case of a Cloud broker reselling VM time from different Cloud providers, the resource allocation goal may aim for instance, to minimize the cost of the resold VMs.

This problem turns out to be similar to the 2-dimensional bin packing problem with rejection. In order to solve this problem, we will use a very straightforward sequential algorithm commonly known as "Decreasing First Fit" (DFF) algorithm. DFF is a simple offline heuristic algorithm that achieves a near-optimal solution for the classical 1-dimensional bin packing problem [Yue91]. The DFF strategy operates in two phases.

First, it sorts the ARs in decreasing order based on their duration $\sum_{\forall j}(\beta_j^i - \alpha_j^i)$. Then, it processes the ARs according to the previous order, and schedules each VM in the first physical host with sufficient remaining capacity during its execution time. If none of the physical hosts can fully accommodate the incoming VM, the AR will be rejected, as previously stated.

5.4.2 Pricing and rewarding end-users

The Cloud provider is responsible for guaranteeing the QoS required by the reservations. In return, the Cloud provider expects the payment of reward or fee for the successful completion of a reservation. If α_j^i denotes the start-time of a VM and β_j^i its expected stop-time estimated by the end-user, the end-user will be charged a fee $\mathcal{F}_j^i$ equal to $(\beta_j^i - \alpha_j^i) \times \Delta^R$, where Δ^R is the hourly rate of a reserved VM. However, it may happen that a VM is needed for more time than initially estimated ($\gamma_j^i > \beta_j^i$). In this case, the Cloud provider can allocate the required resources for a longer period for a higher hourly rate Δ^O on a best-effort basis ($\Delta^O > \Delta^R$). In other words, the Cloud provider cannot guarantee the VM availability until the real stop-time γ_j^i. Let θ_j^i denotes the time when the VM is stopped ($\theta_j^i = \gamma_j^i$) or it is forced to terminate by the Cloud provider if the VM is reserved for executing another end-user ($\theta_j^i < \gamma_j^i$). In the case of under-estimated reservations, the end-user will be charged a fee $\mathcal{F}_j^i$ equal to $(\beta_j^i - \alpha_j^i) \times \Delta^R + (\theta_j^i - \beta_j^i) \times \Delta^O$.

When the Cloud provider accepts an AR, the end-user expects to be able to access the reserved VMs at the specified starting time. However, changes may occur between the time when the end-user submits the reservation and this specified starting time. This can happen for various reasons such as end-users canceling or modifying requests, resource failures, and errors in the estimation of the execution time. Since an AR is a commitment by the Cloud provider, failing to meet this commitment may result in the provider having to pay a penalty $\mathcal{P}_j^i$ to the end-user equal to $(\beta_j^i - \alpha_j^i) \times \Delta^P$.

5.4.3 Resource allocation policies

From the previous discussion, three scenarios have been distinguished: over-estimated ARs (Figure 5.4a), under-estimated ARs without any conflict (Figure 5.4b), and under-estimated ARs resulting in a conflict (Figure 5.4c) with other ARs. The first two scenarios are trivial since the Cloud provider does not have to intervene and the AR will end normally. However, for the third scenario, a Cloud provider motivated by profit has to decide at the arrival of a new AR $\alpha_{j'}^{i+n}$ whether to keep running the under-estimated AR or abort it. In order to tackle this conflictive scenario, we have defined three different resource allocation policies: highest priority to running ARs, highest priority to future ARs and an economic agent.

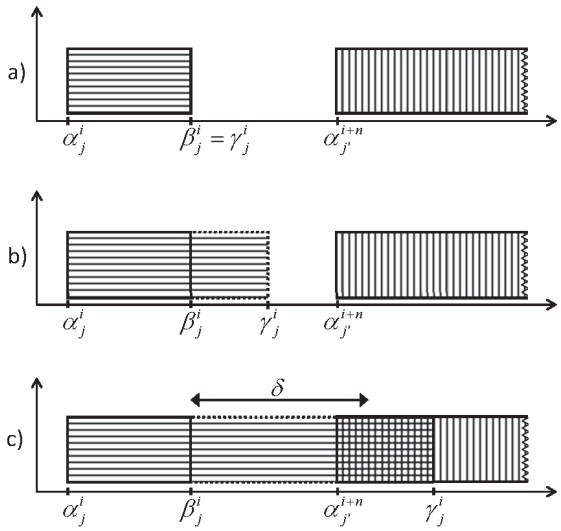

Figure 5.4: Possible scenarios of running Advance Reservations

Strategy 1: Highest priority to running ARs

Under this strategy, the Cloud provider will never stop a running VM and always try to postpone the incoming AR that causes the conflict to a later period through negotiations. The only incentive for the end-user to accurately estimate the time of VM utilization is motivated by the lower price of ARs ($\Delta^O > \Delta^R$). This strategy is characterized by a null percentage of dropped ARs during their execution.

Strategy 2: Highest priority to future ARs

Under this strategy, under-estimated VMs are penalized as they are aborted after they have been started if there is a conflict with a future AR. In order to protect their application from forced termination, end-users with critical applications must ensure that the ARs times are sufficient for their applications to be completed. This strategy is characterized by a null percentage of rejected ARs prior to their execution since all accepted ARs are honored by the Cloud provider.

An economic agent for maximizing revenues under pay-as-you-book

Under this approach, an agent to manage the conflict between currents running under-estimated ARs and future ARs is proposed. The Cloud provider has first to estimate the average extra-time δ required by the currents running under-estimated AR. This can be easily obtained by analyzing the past history of AR executions and hence adjusting δ accordingly. Based on this, the Cloud provider predicts that if the under-estimated AR is kept running, it will get an additional fee of $(\delta + \beta_j^i - \alpha_{j'}^{i+n}) \times \Delta^O$ but will have

to pay a penalty $\mathcal{P}_{j'}^{i+n}$ equal to $(\beta_{j'}^{i+n} - \alpha_{j'}^{i+n}) \times \Delta^P$. If the under-estimated AR is aborted and the new AR is executed, the Cloud provider estimates its gain to be equal to $\mathcal{F}_{j'}^{i+n} = (\beta_{j'}^{i+n} - \alpha_{j'}^{i+n}) \times \Delta^R$. By comparing these two values, the Cloud provider will decide on the way to resolve this conflict. If the Cloud provider decides to keep the under-estimated AR, it should negotiate with the owner of the incoming AR if it accepts to delay its current execution and gets in exchange a penalty and a new time slot for executing its AR. In this study, it is assumed that the end-user can accept such a proposal with a probability ρ.

5.5 Case Study: A Virtual Cloud Provider maximizing revenues through the Pay-as-you-book pricing model

In the mobile business, Mobile Virtual Network Operators (MVNOs) offer attractive mobile communication services without having their own infrastructure or spectrum. As in the mobile business, Cloud brokers may operate in the near future, like Virtual Cloud Providers (VCPs) by assuming credit risk and by creating new pricing models addressing specific market segments. This case study considers a VCP that resells at a flat rate VMs reserved in advance. For this, the VCP reserves across multiple Cloud providers a large number of VMs (quota) at a lower price that the announced by the Cloud providers and resells them under the pay-as-you-book pricing model.

5.5.1 Experimental setup

For our simulations, a VCP with a fixed number $\mathcal{N}$ of same size VMs reserved across multiple Cloud providers is considered. A simulation period of 4 days (or equivalently 96 hours) has been defined. The VCP collects the set of ARs prior to their execution. It is assumed that each AR is composed of a single VM. The start-time α_j^i of a VM is chosen uniformly in the interval $[0, 96)$ while its estimated utilization μ_j^i follows a negative exponential law of mean $\hat{\mu} = 5$ hours bounded by a maximum utilization of 8 hours ($\beta_j^i = \alpha_j^i + \mu_j^i$). The percentage ψ of AR that are under-estimated varies in the set $\{20\%, 30\%, 40\%, 50\%\}$ and the extra-time required by these reservations λ_j^i also follows a negative exponential law of mean $\hat{\lambda}$ equal to 1 or 2 hours ($\gamma_j^i = \beta_j^i + \lambda_j^i$). Without loss of generality, the value of Δ^R has been fixed to 1. Consequently, the parameters Δ^O and Δ^P can take their values in the sets $\{1, 2, 3, 4, 5\}$ and $\{0.5, 1, 1.5\}$, respectively. Finally, the probability ρ of a successful negotiation between the VCP and the end-users was fixed to 100%.

For each simulation, we report the percentage:

- $\mathcal{R}_i$ of ARs that were rejected at the end of the offline initial scheduling;

	$\mathcal{M} = 100$ ARs					$\mathcal{M} = 200$ ARs				
	$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ	$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ
DFF	0.10	0	0	35.45	30 500	7.64	0	0	67.51	58 000
On-Demand	0.20	*	*	38.54	99 750	9.68	*	*	67.01	173 250
Strategy 1	0.10	*	8.17	35.09	33 750	7.64	*	10.01	64.77	61 000
Strategy 2	0.10	8.97	*	37.15	32 750	7.64	11.29	*	69.64	61 000
Economic Agent	0.10	6.42	2.32	37.03	35 250	7.64	8.00	2.91	69.34	64 750

Table 5.2: Impact of the number of submitted Advance Reservations (ARs). *s are 0% by default.

- $\mathcal{R}_d$ of initially accepted ARs that were dropped during their execution because they under-estimated their execution time;
- $\mathcal{R}_r$ of initially accepted ARs that were rejected prior to their execution because the VCP decided to keep running an under-estimated request;
- the percentage $\mathcal{R}_a$ of advanced reservations that are accepted and executed during their complete activity period. It is obvious that the following equation holds:

$$\mathcal{R}_i + \mathcal{R}_d + \mathcal{R}_r + \mathcal{R}_a = 100\% \tag{5.2}$$

as well as:

- the average utilization ratio χ of the VCP resources during the simulation period.
- the revenue Ξ of the VCP computed as a function of Δ^R, Δ^O, and Δ^P;

All the experiments have been repeated 1000 times. The average and the standard deviation computed over these different runs are recorded. In our simulations, it has been considered the three resource allocation policies previously described (*c.f.* Section 5.4.3) as well as the on-demand approach. In the on-demand approach, no ARs are made at all and the resource allocation is performed online. Upon the arrival of a new request, the VCP evaluates its instantaneous resource utilization. If enough free resources are available, the new request is accepted; otherwise, it is rejected. In return, the end-user is expected to pay a higher price Δ^O for accessing the resources as they are not reserved in advance. This approach does not ensure end-user satisfaction with a request for multiple VMs as there is no guarantee that all the VM will be provisioned.

5.5.2 Results and analysis

Impact of the number of submitted ARs

In the first scenario, the parameters of the simulation have been fixed as follows: $\mathcal{N} = 10$, $\Delta^R = 1$, $\Delta^P = 1$, $\Delta^O = 3$, $\psi = 20\%$, $\hat{\lambda} = 1$, $\rho = 100\%$ (Table 5.2).

		$\hat{\lambda} = 1$					$\hat{\lambda} = 2$				
		$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ	$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ
$\psi = 20\%$	DFF	7.64	0	0	67.51	58 000	7.43	0	0	67.58	58 000
	On-Demand	9.68	*	*	67.01	173 250	11.12	*	*	68.31	176 250
	Strategy 1	7.64	*	10.01	64.77	61 000	7.43	*	11.21	65.89	65 000
	Strategy 2	7.64	11.29	*	69.64	61 000	7.43	12.81	*	70.15	61 500
	Economic Agent	7.64	8.00	2.91	69.34	64 750	7.43	5.53	6.51	69.40	67 500
$\psi = 30\%$	DFF	7.44	0	0	67.45	58 000	7.45	0	0	67.49	58 000
	On-Demand	11.10	*	*	68.24	176 250	13.48	*	*	70.66	180 750
	Strategy 1	7.44	*	14.34	63.60	62 000	7.45	*	15.91	65.22	67 750
	Strategy 2	7.44	17.07	*	70.61	62 250	7.45	19.34	*	71.38	63 250
	Economic Agent	7.44	11.89	4.33	70.15	67 750	7.45	9.52	8.09	69.98	71 750
$\psi = 40\%$	DFF	7.51	0	0	67.54	58 000	7.48	0	0	67.48	58 000
	On-Demand	12.68	*	*	69.52	179 250	15.97	*	*	71.27	183 750
	Strategy 1	7.51	*	18.10	62.66	62 750	7.48	*	19.89	64.71	69 750
	Strategy 2	7.51	22.71	*	71.78	63 750	7.48	25.50	*	72.63	62 500
	Economic Agent	7.51	15.67	5.59	71.11	71 000	7.48	12.36	10.27	70.72	75 250

Table 5.3: Impact of the percentage of under-estimated Advance Reservations (ARs) and their execution extra-time. *s are 0% by default.

As expected, the on-demand approach ensures the highest VCP revenue as the end-users are paying a higher price during all the execution of their tasks ($\Delta^O = 3 \times \Delta^R$). It also achieves a high overall acceptance ratio $\mathcal{R}_a$ as it does not have to deal with estimation uncertainties. We notice that both strategies 1 and 2 achieve similar revenue Ξ for the VCP. However, Strategy 1 achieves the highest acceptance ratio $\mathcal{R}_a$ for AR, while Strategy 2 has a better performance in terms of resource utilization χ. Our proposed economic agent achieves slightly lower resources utilization compared to Strategy 2 and keeps the percentage of rejected AR prior to their execution $\mathcal{R}_d$ at an acceptable value. In summary, our proposed economic agent is a trade-off in terms of resource utilization and acceptance ratio between the intuitive strategies 1 and 2, but outperforms both of them in terms of VCP revenue. These conclusions hold independently of the number of submitted ARs.

Impact of the percentage under-estimated ARs and their execution extra-time

In the second scenario, the parameters of the simulation have been fixed as follows: $N = 10$, $\Delta^R = 1$, $\Delta^P = 1$, $\Delta^O = 3$, $\mathcal{M} = 200$, $\rho = 100\%$ (Table 5.3).

As the initial scheduling does not have any knowledge about the error in estimating the execution time, it achieves the same performance independently of the values of ψ and $\hat{\lambda}$. As the percentage of under-estimated ARs increases, the percentage of ARs that are rejected prior to their execution in Strategy 1 increases also. However, this increase is less pronounced than the increase observed in the Strategy 2 for the percentage of dropped ARs during their execution. Finally, our proposed economic agent keeps its superiority

and still achieves a trade-off in terms of resource utilization and acceptance ratio between the strategies 1 and 2, but it outperforms both of them in terms of VCP revenue.

In general, the results show that the on-demand approach is better in terms of revenues than the proposed economic agent, and the strategies 1 and 2. Since the main interest for a Cloud provider is to maximize its revenues, the obtained results explain why a pricing policy such as pay-as-you-book has not been implemented by Cloud providers. Thus, pay-as-you-book may be implemented by a Cloud broker taking advantage of multiple Cloud providers' service offerings, acting as a VCP.

5.6 Summary

In this chapter the problem of resource provisioning while assuming AR under the *pay-as-you-book* pricing model has been investigated. The proposed model handles the extra-time required by running ARs at a higher price, on a best-effort basis. Indeed, an extra-time of an AR plan may lead to resource conflicts with other AR plans. In order to resolve such resource conflicts, an economic agent responsible for managing the under-provisioning problem has been proposed. The economic agent aims to achieve the provider satisfaction by maximizing its revenues through intelligent resource management. In order to assess the performance of the proposed agent, the proposed economic agent has been compared with two intuitive approaches that systematically prioritize reserved ARs or currently running ARs. The economic agent achieves a trade-off between the two intuitive strategies in terms of resource utilization and acceptance ratio, while outperforming both in terms of provider's revenue. These conclusions hold independently of the number of submitted ARs, the percentage of under-estimated ARs, and the average duration of the extra-time required.

Chapter 6

Conclusion and future works

The aim of this dissertation has been to propose new value-added services and pricing models in Cloud brokering at the infrastructure level. With this aim in view, Chapter 1 provided a comprehensive overview of the current and future value-added services in Cloud brokering. After surveying the research related to Cloud performance evaluation and placement in Cloud brokering (Chapter 2), needs and shortcomings in the current Cloud computing service offerings have been identified. In particular, in the first part of this dissertation (Chapters 3 and 4), the problem of a single figure of merit for Cloud performance and the problem of VM placement in Cloud brokering have been addressed. In the second part of this dissertation, a new pricing model for Cloud computing known as pay-as-you-book has been proposed (Chapters 5).

The computation of a single figure of merit of VM Cloud performance has been described as a multi-criteria problem (Chapter 3). This problem relies on eight criteria: Communication, Computation, Memory, Storage, Availability, Reliability, Scalability and Variability (Section 2.2.3). The weight of these criteria in the computation of a figure of merit of Cloud performance depends on the application profile foreseen to run on top of the Cloud infrastructure. The Analytic Hierarchy Process (AHP) has been used to analyze and to solve the Multiple Criteria Decision Making (MCDM) problem of finding a single figure of merit of Cloud performance. In this case, AHP enables an objective determination of the relative merit of the VM performance criteria for a given set of Cloud providers.

Similarly to the problem of finding a figure of merit of Cloud performance, the problem of placement in Cloud brokering has been described as a multi-criteria problem (Chapter 4). This problem refers to the efficient distribution of Cloud infrastructure across multiple and non-interoperable Cloud providers. Preemptive goal programming has been used to tackle this problem by defining a set of multiple LPs with different priorities assigned by the end-user.

A pricing model between pay-as-you-go and subscription-based known as pay-as-you-book has been proposed (Chapter 5). Contrary to subscription-based pricing models, pay-as-you-book allows reservations of Cloud resources for future use without long-term commitment. Three resource allocation policies to manage the extra-time required by running reservations under pay-as-you-book have been described and evaluated. Among the evaluated policies, the economic agent maximizes Cloud provider's revenue while keeping an acceptable ratio of resource utilization.

Cloud brokers have emerged in the Cloud computing landscape as a technical solution to bring unified self-service access to multiple non-interoperable Cloud providers. By bringing interoperability and portability of end-user's applications across multiple Cloud providers, Cloud brokers act as an ideal doorway to fill the current technical gaps in Cloud computing and to introduce new pricing and business models. Technically, Cloud brokers already complement or enhance some of the Cloud provider service offerings such as infrastructure monitoring, cost optimization, elasticity management and consolidated billing. In this manner, Cloud brokers act as a single point of access for consumption of Cloud services. With the introduction of new value-added services, as those exposed in this dissertation, Cloud brokers may become trusted third-parties, providing un-biased information that benefits end-users.

Current Cloud providers look for differentiation through the addition of new value-added services to their portfolios. Similarly to supermarkets, Cloud providers become a place to find aggregated but non-interoperable services. For example, Cloud providers do not provide infrastructure monitoring services that monitor the infrastructure of their competitors. In the near future, Cloud brokers can take part of the Cloud economies by actively changing the value chain of Cloud computing. Similarly to the Mobile Virtual Network Operator (MVNO) business model, Cloud brokers, without any hardware infrastructure, may develop new and appealing pricing models addressing untapped market segments. By acting as a single point of access for consumption of Cloud services, Cloud brokers could set the bar of how much end-users should pay for a given Cloud offer depending on the SLA of the Cloud provider and its respective performance. Thereby, Cloud brokers will increase competition between Cloud providers.

The practical implication of this dissertation is threefold. First, the proposed figure of merit can be used to objectively compare Cloud providers based on their performance and on the application profile to be deployed. Second, the computation of this figure of merit linked with the proposed intelligence for Cloud brokering placement optimizes costs of the distributed resources depending on the end-user constraints. This intelligence may enrich the service portfolio of not only Cloud brokers, who could automatically respond to unforeseen scenarios, but also consultancy firms and IT departments who may take data-driven decisions when migrating into the Cloud. Third, the proposed pricing model is a first step to the study of mechanisms enabling new and appealing ways of purchasing

Cloud infrastructure.

This work has identified two areas for possible further study. These include the identification of standard sizes and the establishment of standard SLAs for Cloud VMs. The definition of standard VM sizes solves the problem of product differentiation created by current heterogeneous VM service offerings from Cloud providers. Thus, the challenge is to identify a measure of VM configurations which satisfies the largest demand of end-users by taking into account the different application profiles. Cloud SLAs vary from one Cloud provider to another. In order to enable the comparison of service offerings, SLAs terms and definitions need to be standardized across Cloud providers. In summary, standard SLAs, standard VM sizes along with our proposed figure of merit contribute to the commoditization of Cloud VMs.

Appendix A

Cloud performance evaluation: details and extended results

Contents

A.1 Related issues to the performance evaluation

Issues faced during the development of this study:

- Not all Cloud providers provide an API to manage the VMs. This fact obliged us to start and stop VMs via the web interface which prevents the exact measurement of the provisioning time.
- Some Cloud providers (particularly the recently emerged) do not support the import of VM images.
- The image provided by one Cloud provider had the root user account deactivated (for security reasons as expressed by the technical support). As *ceilo* was conceived for being used under the root account, we faced some troubles at the moment of installing and configuring the benchmarks.

- Acquisition of Cloud resources is not automatic for all Cloud providers. For instance, for some Cloud providers, the creation of the account needed to be validated by human-intervention before we cloud use the resources. Sometimes the confirmation took more than one working day.
- One Cloud provider had a security policy that considered our benchmarks a risk for its Cloud infrastructure. Our VMs were immediately stopped and the account got blocked till we explained the reason behind our tests.
- In some cases, the online documentation is extensive and well-explained, in some other cases the documentation is insufficient to solve technical issues but the technical support assisted us in deploying the applications.

A.2 VM configurations

The table A.1 presents the evaluated VM configurations. All prices have been converted to US$ (1US$ = 1.23€).

Table A.1: VM configurations

Cloud provider	VM size	vCPU (number)	RAM (GB)	Disk (GB)	Price(US$)/hr
Arubacloud	s	1	2	10	0.0309
	m	2	4	20	0.0556
	l	4	8	40	0.1050
Amazon	xs	1	0.615	8	0.0200
	s	1	1.7	160	0.0650
	m	1	3.75	410	0.1300
	l	2	7.5	840	0.2600
	xl	4	15	1680	0.5200
Cloudsigma	xs	1	0.512	10	0.0524
	s	1	2	10	0.0807
	m	2	4	10	0.1526
	l	4	8	10	0.2339
	xl	8	16	10	0.5841
Joyent	xs	0.15	0.625	20	0.0200
	s	1	1.75	56	0.0560
	m	2	7.5	738	0.2400
	l	4	15	1467	0.4800
	xl	8	30	1683	0.9600
Lunacloud	xs	1	0.512	10	0.0191
	s	1	2	10	0.0469

Continued on next page

Table A.1: (Continued)

Cloud provider	VM size	vCPU (number)	RAM (GB)	Disk (GB)	Price(U$)/hr
	m	2	4	20	0.0939
	l	4	8	40	0.1870
	xl	8	16	80	0.3750
Profitbricks	s	1	2	10	0.0413
	m	2	4	20	0.0825
	l	4	8	40	0.1650
	xl	8	16	80	0.3300
Rackspace	xs	1	0.512	20	0.0330
	s	1	1	40	0.1210
	m	2	4	160	0.2430
	l	4	8	320	0.4870
	xl	8	30	1200	1.5240
WindowsAzure	xs	2-shared	0.768	20	0.0184
	s	1	1.75	70	0.0552
	m	2	3.5	135	0.1105
	l	4	7	285	0.2210
	xl	8	14	605	0.4410

A.3 Benchmark duration

The benchmark duration is important when measuring Cloud performance, since the costs related with the evaluation are directly proportional to its duration. Regarding the benchmark duration (Figure A.1), half of the Cloud providers (Arubacloud, Cloudsigma, Profitbricks and Rackspace) have a benchmark duration under one hour for all VM sizes. For the others, the benchmark duration and VM size are inversely proportional. This proportional relationship is mainly due to three facts. First, we kept a constant workload in the computation benchmarks (*7zip* and *c-ray*) across all VM sizes. Thus, the lower the number of processors in a VM, the longer the duration of the computation benchmarks. Second, the practice of processor-sharing by Cloud providers increases the benchmark duration. For example, Amazon and Joyent share the processor time between VMs for *xs*-VMs. Third, the differences in processor brands and qualities make some Cloud providers more powerful than others in computing terms (Table A.2).

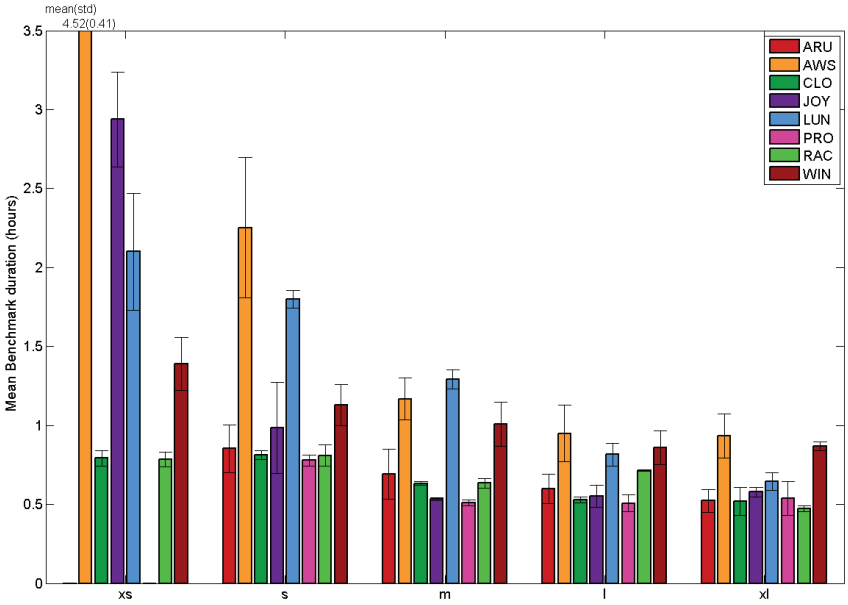

Figure A.1: Benchmark duration

Provider	Processor
AWS	Intel Xeon E5-2650 @ 1.80GHz
CLO	AMD Opteron 6380 @ 2.50GHz
JOY	Intel Xeon E5645 @ 2.40GHz
LUN	Intel Xeon E5-2620 @ 1.50GHz
RAC	AMD Opteron 4332 HE @ 3.00GHz
WIN	AMD Opteron 4171 HE @ 2.09GHz

Table A.2: Type of processor for *xs*-VM size

A.4 Performance-price correlation with a simple figure of merit of Cloud performance

The results here presented were calculated with the simple figure of merit of Cloud performance method (c.f. Section 3.3.2).

A.4.1 Correlation among VM sizes from different Cloud providers

The performance-price relationship for the same VM size from different Cloud providers has been studied. The performance values have been calculated as previously described for each VM size (*i.e.* for every graph presented in Figure A.2). The Average Values (AVE) have been used to split each graph into four quadrants: High Performance (HP) and Low Cost (LC), High Performance (HP) and High Cost (HC), Low Performance (LP) and Low Cost (LC), and, Low Performance (LP) and High Cost (HC). The highlights of the findings are the following:

- Arubacloud presents the best performance-price relationship among the evaluated Cloud providers for the three VM sizes (s,m and l) evaluated with a low variability in the case of s and m sizes.
- AWS is placed on the HC-LP and LC-LP quadrants for all the VMs sizes but for the m VM size. AWS VMs present a low variability (0-10%) for the m and xl sizes.
- Cloudsigma presents a HP and a small variability at a HC for the small sizes (xs and s). For the m, l and xl sizes the performance is close to the AVE. VMs have a low variability for all the sizes but the m size.
- Joyent has a HP for all the VMs sizes (but xs size) at a HC for the m, l and xl sizes. VMs have a low variability for all the sizes but the xl size.
- Lunacloud VMs are on the HP-LC and LP-LC quadrants. VMs have a performance over the AVE for the l and xl sizes, with a low variability for the m and l sizes.
- Profitbricks VMs are on the HP-LC quadrant. VMs have a low variability for the s, m and l sizes.
- Rackspace presents a low variability and is placed on the HP-HC and HP-LC quadrants for all the VMs sizes.
- WindowsAzure VMs are on the HP-LC and LP-LC quadrants. Performance results are 1 point over the AVE values for xs and xl sizes and 1 point under for s, l and m sizes. In cost WindowsAzure is under the AVE for all VMs sizes. Low variability is presented in m, l and xl VMs.

A.4.2 Correlation among different VM sizes from a single Cloud provider

The performance-price relationship for different VM sizes from the same Cloud provider has been studied here. The performance values have been calculated as previously described. The motivation behind this is to check the correspondence among size, price and performance of VMs. In general, prices are proportional to the size and performance of the VMs (Figure A.3). The highlights of our findings are the following:

Economic advantage

A VM-pair comparison for every Cloud provider in order to find VMs with similar performance values has been made. Users may reduce costs by using cheaper VMs with equivalent performance. For each Cloud provider, pairs of VMs ($\mathrm{VM}_x, \mathrm{VM}_y$) have been

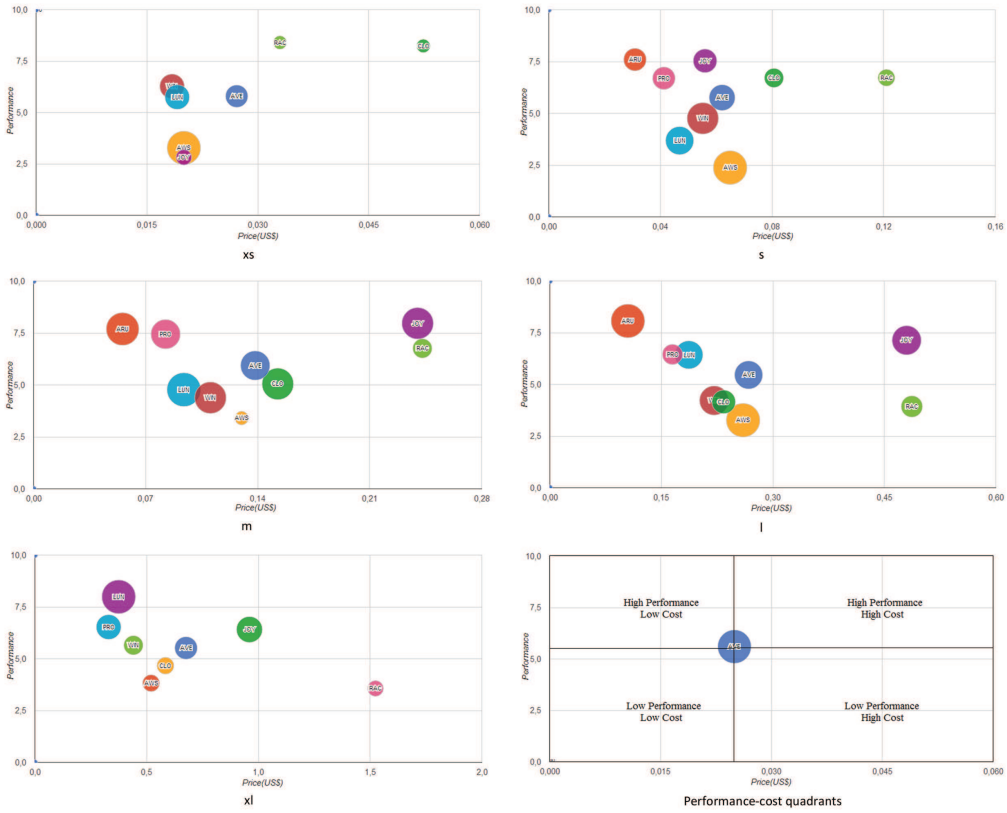

Figure A.2: Correlation between performance and price for VM sizes. The variability of a VM is represented by the size of the spot. Lower bound equal to 1 and upper bound equal to 10.

found as follows. For two VMs, VM_x and VM_y, where:

$$Size(\mathrm{VM}_x) < Size(\mathrm{VM}_y) \tag{A.1}$$

The pair $(\mathrm{VM}_x, \mathrm{VM}_y)$ is selected if:

$$Performance(\mathrm{VM}_x) \geq Performance(\mathrm{VM}_y) \tag{A.2}$$

or if:

$$0 < Performance(\mathrm{VM}_y) - Performance(\mathrm{VM}_x) < 0.5 \tag{A.3}$$

We define the Economic Advantage (EA) as the amount of money a user saved when choosing the smallest VM between two VMs with similar performance. EA is denoted as

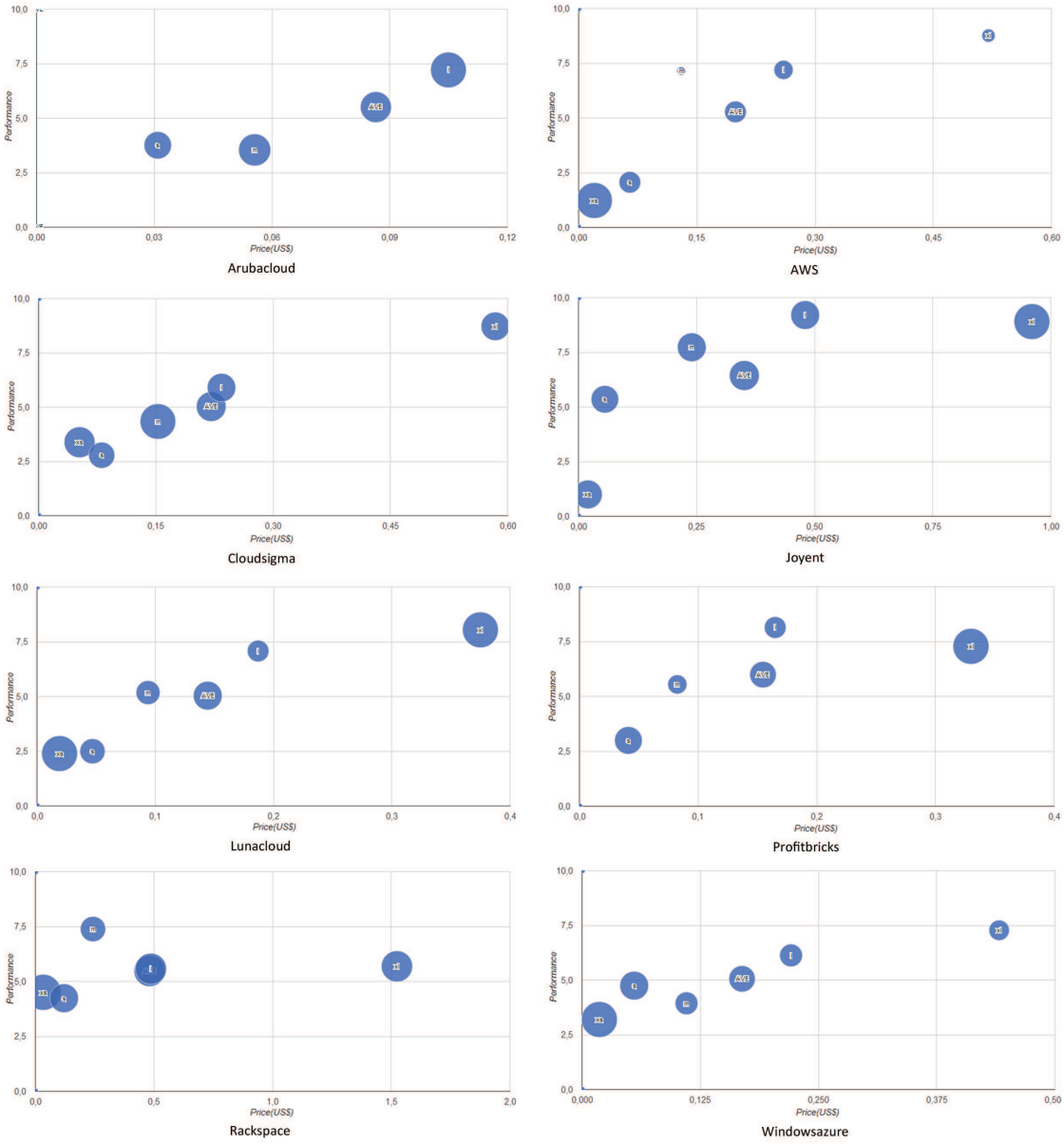

Figure A.3: Correlation between performance and price for Cloud providers. The variability of a VM is represented by the size of the spot. Lower bound equal to 1 and upper bound equal to 10.

follows:

$$\text{EA} = \left[1 - \frac{Price(\text{VM}_x)}{Price(\text{VM}_y)}\right] \times 100\% \tag{A.4}$$

Table A.3 presents the pairs of VMs with their correspondent EA that satisfy equation (A.2) or (A.3).

Provider	($\mathbf{VM}_x, \mathbf{VM}_y$)	EA
ARU	(s,m)	50%
AWS	(m,l)	50%
CLO	(xs,s)	37.5%
JOY	(l,xl)	50%
LUN	(xs,s)	60%
PRO	(l,xl)	49.5%
RAC	(xs,s)	25%
	(m,l)	51%
	(l,xl)	67.8%
WIN	(s,m)	45.5%

Table A.3: VM-pair and EA

Appendix B

Résumé en français

Contents

B.1 Introduction

Le rôle des « *Cloud brokers* » dans l'avenir du « *Cloud computing* » a été présenté par Gartner comme une tendance majeure pour les prochaines années : « *Pour l'année 2015, les Cloud brokers représenteront l'unique et la plus grande catégorie de croissance en Cloud computing, passant d'un marché d'environ un milliard de dollars américains en 2010 à un marché de plusieurs centaines de milliards de dollars.* » [Can12]. Cette prédiction est

confirmée par le montant des fonds lévés par quelques entreprises de « *Cloud brokering* » : *RighScale* a obtenu 47.3 millions de dollars en trois levées de fonds[1], *6fusion* 10 millions de dollars en deux levées de fonds, *Cloud Cruiser* 7.6 millions de dollars en deux levées de fonds, *Zimory Systems* 7.2 millions de dollars en deux levées de fonds et *Gravitant* 3.7 millions de dollars en une levée de fonds [Fel13]. L'une des principales raisons de cette forte attente économique est la forte hétérogénéité du marché actuel du Cloud constitué par plusieurs fournisseurs de service. Dans ce marché, chaque fournisseur propose ses interfaces, ses modèles de prix et ses services à valeur ajoutée. Afin d'aider les consommateurs de Cloud à faire face à un tel écosystème aussi fragmenté, les Cloud brokers sont devenus des intermédiaires qui fournissent un accès unique à plusieurs fournisseurs de Cloud. Ainsi, les Cloud brokers offrent un unique point d'accès pour la consommation de services, permettant ainsi l'interopérabilité et la portabilité des applications à travers de multiples fournisseurs de Cloud.

Parmi les autres services à valeur ajoutée fournis par les Cloud brokers (Figure B.1), on trouve : la gestion avancée des offres Cloud grâce à des outils plus performants que de ceux déjà proposés par les fournisseurs ; la gestion de l'élasticité permettant d'augmenter ou de diminuer automatiquement les ressources en infrastructure pour le Cloud ; le choix optimal entre plusieurs services. Ces services permettent la création de nouveaux scénarios avantageux pour les consommateurs, ainsi que pour les fournisseurs. Dans le cas de « *Cloudbursting* », les consommateurs ont la possibilité d'étendre leurs installations informatiques en développant leurs applications non critiques chez les fournisseurs publics. Dans le cas des places de marché, les consommateurs ont accès à des fournisseurs multiples à travers une interface unique, tandis que les fournisseurs ont la possibilité de louer l'infrastructure inutilisée.

Les Cloud brokers pourraient favoriser la création de valeur à travers des services avancés à forte valeur ajoutée, permettant l'émergence de nouveaux cas d'usage pour le Cloud computing. Les prix des infrastructures de Cloud computing varient autour de 20% selon les fournisseurs tandis que les différences de performance entre fournisseurs restent inconnus ou moins étudiées [Fel13]. Comme les Cloud brokers sont en mesure de déployer une charge de travail chez n'importe quel fournisseur, la mesure de la performance et l'allocation des ressources en infrastructure basée sur un compromis coût/performance constitueront de nouveaux services à valeur ajoutée de la part des brokers. D'autre part, la création d'une unité de valeur pour évaluer les performance des infrastructure Cloud contribue à faire du Cloud une utilité publique, ce qui augmentera l'adoption du Cloud computing par le marché et simplifiera l'achat de ressources. Ainsi, si l'infrastructure Cloud est négociée comme toute autre utilité publique (par ex. l'eau, l'électricité), le marché jusqu'à présent fragmenté par des offres hétérogènes sera consolidé. Cela permettra de nouvelles politiques de prix où les brokers serviront non seulement d'intermédiaires,

[1] Une levée de fonds est une pratique par laquelle une entreprise lève des fonds pour financer son expansion, une acquisition ou dans un autre but.

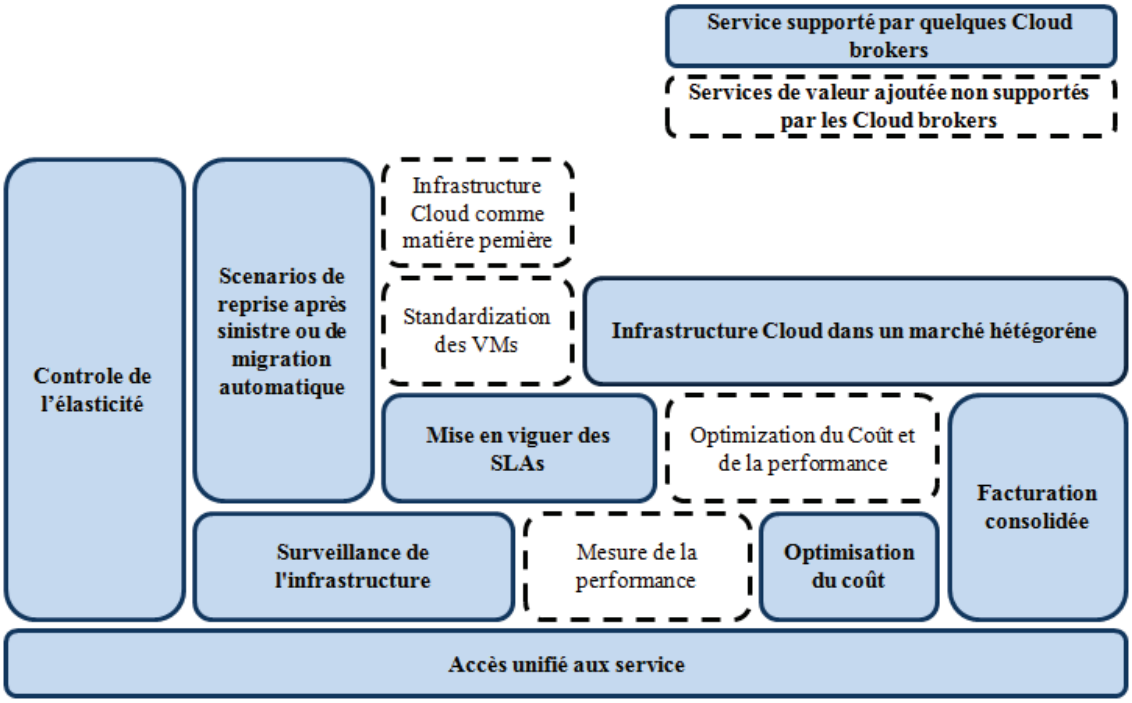

FIGURE B.1 – Evolution des services de valeur ajoutée dans le Cloud brokering

mais aussi de fournisseurs de liquidité négociant des réductions de prix et qui garantissant aux consommateurs la disponibilité de la ressource.

Cette thèse se concentre sur les services à valeur ajoutée et les politiques de prix d'un point de vue de l'infrastructure Cloud. Trois objectifs sont visés : le premier est de proposer une mesure de qualité de la performance des machines virtuelles basé sur le profil d'application. Le deuxième est de proposer une approche exacte pour l'allocation de machines virtuelles à travers de plusieurs fournisseurs basée sur différents critères d'optimisation. En fin, le troisième est de décrire une politique de prix pour le Cloud brokering, appelée « *pay-as-you-book* ».

B.2 Mesures de performances des fournisseurs de Cloud

B.2.1 Enjeux

Actuellement, il est impossible de faire une comparaison directe des offres de services Cloud. Dans le cas des ressources en infrastructure, cela est dû principalement à l'hétérogénéité des configurations des VMs. D'une part, les fournisseurs de Cloud traditionnels tels qu'Amazon, Rackspace et WindowsAzure vendent des machines virtuelles de taille fixe (c'est-à-dire des machines virtuelles avec une configuration prédéfinie). Ces configurations des machines virtuelles varient parmi les fournisseurs, il n'est donc pas possible de trou-

ver la même configuration de machine virtuelle chez deux fournisseurs différents. D'autre part, les nouveaux fournisseurs de Cloud, afin d'augmenter leur attractivité auprès du consommateur, cherchent à différencier leurs services en permettant aux consommateurs de configurer la quantité des ressources d'infrastructure qui leur est nécessaire.

L'évaluation de la performance des machines virtuelles augmente la complexité de la comparaison des fournisseurs de Cloud. Tout d'abord, les consommateurs ont peu de connaissances et peu de contrôle sur l'infrastructure sur laquelle sont hébergées leurs applications. En raison de la virtualisation du matériel utilisé par les fournisseurs de Cloud, les fournisseurs peuvent favoriser le partage de ressources (par exemple le partage du processeur par plusieurs processus, la surréservation de la mémoire, l'étranglement ou sous-dimensionnement du réseau [HZKD11]) qui dégradent les performances d'une application exécutée sur le Cloud. D'autre part, les centres de données des fournisseurs sont équipés avec des centaines de milliers de serveurs avec une qualités matérielle et logiciel variable. Ainsi, l'évaluation de la performance, de tous les centres de données chez plusieurs fournisseurs de Cloud, suppose un compromis entre la précision, le temps et le coût de l'évaluation [LYKZ10]. Par ailleurs, les fournisseurs mettent régulièrement à jour ou étendent leurs infrastructures matérielles et logicielles, de nouvelles technologies et des services commerciaux pouvant progressivement entrer sur le marché [LZO+13]. Par conséquent, les évaluations de la performance deviennent rapidement obsolètes et les outils de mesure de performance doivent être continuellement mis à jour. Enfin, il n'y a pas de logiciel de référence spécialisé pour évaluer de manière global les caractéristiques des machines virtuelles [IPE12]. Cependant, les logiciels traditionnels peuvent satisfaire partiellement les exigences d'évaluation de la performance.

L'évaluation de la performance des machines virtuelles serait bénéfique pour les consommateurs et les fournisseurs de Cloud [LYKZ10]. En effet, celle-ci permettrait aux consommateurs de tester leurs applications chez de multiples fournisseurs et ainsi de choisir le fournisseur qui représente le meilleur rapport performance/coût. En outre, les évaluations peuvent servir de recommandation de la performance d'un système particulier [HZKD11] ou peuvent donner aux consommateurs des arguments techniques pour inciter les fournisseurs de Cloud à metre en oeuvre les meilleures pratiques en matière d'infrastructure [IPE12]. Aussi, les évaluations permettent à un fournisseur d'identifier son positionnement sur le marché afin d'améliorer ses services ou de modifier ses prix [LYKZ10].

B.2.2 Études relatives à l'évaluation de la performance des services de Cloud

Une étude exhaustive des approches académiques d'évaluation des services commerciaux de Cloud a été réalisée par l'Université National d'Australie [LZO+13]. Une « *Systematic Literature Review* » (SLR) fut la méthode utilisée pour recueillir les données pertinentes

pour rechercher l'évaluation des services Cloud. 82 études d'évaluation des services de Cloud ont été relevées. Les principales conclusions de cette étude représentent un état de l'art en ce qui concerne l'évaluation de services Cloud. Ces conclusions sont les suivantes :

- 50% des études cherchent à appliquer le Cloud computing aux problèmes scientifiques, tandis que seulement 16% des études se penchent l'évaluation des applications pour les entreprise dans le Cloud.
- 21 services de Cloud ont été sélectionnés chez 9 fournisseurs de Cloud. 70% des études évaluent les services fournis par Amazon Web Services uniquement.
- Trois aspects principaux de l'évaluation de la performance des services Cloud avec leurs propriétés respectives ont été étudiés : la performance, l'économie et la sécurité. La performance étant l'aspect le plus étudié (78 études).
- Il n'y a pas de consensus sur la définition et le type des métriques utilisées. Certaines métriques de même nom ont été utilisées pour designer différentes mesures. De la même façon, certaines métriques avec différents noms correspondent à une même mesure. L'étude a identifié plus de 500 métriques, y compris les doublons.
- Il y a un manque de métriques efficaces vis-à-vis de l'élasticité et des aspects de sécurité en Cloud computing. En conséquence, il n'est pas possible de quantifier la quanlité d'élasticité et de sécurité d'un service Cloud.
- Il n'y a pas de logiciels de référence fournissant une évaluation globale de services du Cloud. Le SLR a recensé environ 90 logiciels différents utilisés dans les études sur l'évaluation des services Cloud. Ces logiciels de référence ont été regroupés en trois catégories principales : logiciels d'application, micro-logiciels et logiciels à charge de travail synthétiques ils seront expliqués ci-dessous.
- 25 scénarios de base pour l'élaboration de services d'évaluation des services de Cloud ont été identifiés et classés.
- L'évaluation des services de Cloud est de plus en plus étudiée par la communauté scientifique. Le nombre d'études relevées a été multiplié par 17 fois entre 2007 (2 études) et 2011 (34 études).

L'évaluation de la performance des services de Cloud est faite à l'aide de logiciels d'application, de logiciels à charge de travail synthétique et de micro-logiciels. Les logiciels d'application correspondent aux logiciels utilisés dans les environnements de production et fournissent une vue d'ensemble de la performance d'une application spécifique. Les logiciels à charge de travail synthétique simulent le comportement d'une application en imposant une charge de travail sur le système. De même, les micro-logiciels imposent une charge de travail dans le but de mesurer les ressources matérielles spécifiques qui caractérisent une machine virtuelle. Comme il n'existe d'ensemble de logiciel de références spécifique pour l'évaluation des services de Cloud, la performance des ressources

en infrastructure Cloud a été mesurée à l'aide de logiciels tels que TPC-W (logiciel d'*e-Commerce*) [LOCZ12], HPCC (ensemble de sept logiciels pour le calcul haute performance) [SASA+11, IOY+11a, HZKD11], NPB (ensemble de logiciels pour évaluer la performance des super-ordinateurs en parallèle) [MVML11, HZKD11] ou des outils de mesure communs tels que *ping* ou *iperf* [SDQR10, BK10]. En outre, des logiciels spécialisés ont été développés pour mesurer la performance en puissance de calcul, en mémoire vive, les performance du disque et du réseau [ADWC10, HLM+10] ainsi que le temps d'approvisionnement ou de libération des machines virtuelles [SDQR10, IOY+11a, MH12] (davantage de détails sur les études relatives à l'évaluation de la performance des fournisseurs sont présentés dans le tableau 2.1).

B.2.3 Caractérisation des machines virtuelles

Selon les études d'évaluation de la performance des fournisseurs présentées dans la section ci-dessus, une machine virtuelle peut être représentée par un ensemble de critères et un ensemble de métriques. Les critères caractérisent la machine virtuelle (par exemple la communication, la puissance de calcul, la mémoire vive et le stockage) et caractérisent le fonctionnement de machines virtuelles (par exemple la disponibilité, la fiabilité, l'élasticité et la variabilité). L'ensemble des métriques correspond aux métriques utilisées pour décrire la validité des critères. Une courte description des critères et des métriques est présentée ci-dessous.

Critères

- *La communication* est la transmission de données entre deux entités à travers un réseau. On distingue trois types de communication : via un réseau intenrne à un centre de données, via un réseau entre centres de données et via un réseau étendu (WAN, Wide Area Network). Le réseau interne à un centre de données fait communiquer les machines virtuelles appartenant à un même centre de données, tandis que le réseau entre centres de données fait communiquer les centres de données appartenant à un même fournisseur de Cloud. Le réseau étendu fait référence aux communications entre une machine virtuelle allouée dans un centre de données et une machine connectée à Internet.
- *La puissance de calcul* correspond à la performance du processeur pour le traitement de données.
- *La mémoire vive* fait référence à la propriété physique pour la sauvegarde temporaire de données. On considère à la fois la mémoire RAM et et la mémoire cache.
- *Le stockage* fait référence à la propriété physique pour la sauve garde permanente de données.

- *La disponibilité* correspond au pourcentage du temps pendant lequel un utilisateur peut accéder à un service Cloud. Pour un intervalle de temps donné, la disponibilité est calculée comme le rapport entre le temps où le service est disponible et l'intervalle de temps total ; elle est généralement exprimée sur la base d'une année complète.
- *La fiabilité* est la caratéristique d'un service Cloud opérationnel pour une période de temps spécifique.
- *L'élasticité* est la rapidité d'adaptation des capacités de services Cloud en fonction de la demande de consommateurs [ILFL12]. On distingue deux types d'élasticités : *Horizontale* [VRMB11, WCC12] et *Verticale* [DTM11, YF12]. La première fait référence à l'approvisionnement de multiples instances d'un service Cloud (par exemple le déploiement d'une nouvelle machine virtuelle). La deuxième correspond à l'ajout de ressources aux services Cloud déjà déployés (par exemple ajout dynamique de processeurs ou de stockage sur une machine virtuelle déjà existante).
- *La variabilité* est une métrique dérivée d'autres métriques qui fait référence à la variation de la performance d'un service Cloud.

Métriques

Li *et al.* [LOCZ12] définissent les métriques suivantes :

- *La vitesse de transaction* est définie comme le nombre de transactions (par exemple l'exécution d'une tâche, la lecture ou l'écriture en mémoire) traitées par unité de temps.
- *Le débit de données (la bande passante)* est la quantité de données traitées dans une période de temps donnée.
- *La latence* regroupe toutes les métriques du temps d'un service Cloud.
- *Les autres métriques* sont composées de paramètres sans dimension (par exemple la disponibilité) ou de mesures simples telles que la fiabilité.

B.2.4 Mesure de performance Cloud

La moyenne et le graphique radar

La plupart des études liées sur l'évaluation de la performance en *cloud* computing n'utilisent que des mesures de performance indépendantes (Tableau 2.1). Par ailleurs, Li *et al.* [LOZC13] ont proposé une solution pour exprimer la performance globale d'un service de Cloud par un score unique. Cette approche propose la *moyenne* et le *graphique radar* comme méthodologies pour calculer un score unique exprimant la performance d'une

infrastructure Cloud. Cependant, le calcul d'une moyenne de résultats obtenus à l'aide de différents logiciels de référence a le désavantage de nécessiter l'utilisation de la même métrique pour tous les logiciels. Cet inconvénient est surmonté en utilisant le graphique radar ; un graphique radar est un outil permettant de représenter graphiquement trois, ou davantage de valeurs quantitatives et relatives à un point commun. Deux méthodes de normalisation sont proposés par Li *et. al* lorsque les résultats des logiciels de référence sont exprimés dans des unités de valeur différentes : le plus grand et meilleur (HB, abrégé en anglais) (Equation B.1) et le plus petit et meilleur (LB, abrégé en anglais) (Equation B.2).

$$\text{HB Normalisé}_i = \frac{\text{Logiciel de référence}_i}{\text{MAX}(\text{Logiciel de référence}_{1,\dots,n})} \tag{B.1}$$

$$\text{LB Normalisé}_i = \frac{\frac{1}{\text{Logiciel de référence}_i}}{\text{MAX}(\frac{1}{\text{Logiciel de référence}_{1,\dots,n}})} \tag{B.2}$$

Où HB Normalisé$_i$ et LB Normalisé$_i$ correspondent au résultat du $i$ème logiciel de référence. De cette façon, la surface du polygone représentant n résultats peut être considérée comme un score unique de la performance de l'infrastructure Cloud (Equation B.3) [LOZC13].

$$\text{Score unique}_{(\text{graphique radar})} = \sum_{i=1}^{n} \frac{\sin\left(\frac{2\pi}{n}\right) \times \text{Normalisé}_i \times \text{Normalisé}_{mod(i+1,n)}}{2} \tag{B.3}$$

Bien que la moyenne et le graphique radar permettent d'exprimer un score unique, ils présentent des désavantages tels que le manque de pondération et DE nombre fini de valeurs possibles.

Réduction à une échelle commune entre deux bornes

La réduction à une échelle commune entre deux bornes est une méthode employée par les entreprises qui mesurent la performance de l'infrastructure Cloud tels que CloudSpectator et Cloudharmony. Dans cette approche, chaque résultat du logiciel de référence est ramené entre deux valeurs fixes A et B, où A correspond à la borne inférieure (c'est-à-dire le moins bon résultat de performance) et B correspond à la borne supérieure (c'est-à-dire le meilleur résultat de performance). Les valeurs intermédiaires sont ramenées à une échelle entre ces deux bornes.

Critères	Capacités	Logiciels de référence	Métrique	Type
Puissance de calcul	Vitesse de transaction	7zip [zip]	MIPS[2]	HB
		C-ray [cra]	secondes	LB
Mémoire	Débit de données	Stream [Str]	MB/s	HB
		CacheBench [Cac]	MB/s	HB
Stockage	Vitesse de transaction	Threaded I/O Tester [TIO]	MB/s	HB
		Iozone [ioz]	MB/s	HB

Table B.1 – Logiciels de référence

La méthode de hiérarchie multicritère

Dans le cadre de cette thèse, la méthode de hiérarchie multicritère ou Analytic Hierarchy Process (AHP) de Saaty a été employée pour calculer un score unique de performance des machines virtuelles. AHP est une technique pour analyser, organiser et résoudre les problèmes de prise de décision à plusieurs critères [Saa80]. Dans AHP, les problèmes complexes sont simplifiés et structurés en organisant les critères de décision dans une structure hiérarchique. Les compromis entre les critères sont déterminés en faisant des comparaisons par paires. A la différence des méthodes traditionnelles, AHP est basé sur des mesures d'évaluation subjectives et objectives. Dans le domaine du Cloud computing, AHP a été utilisé pour classer les services Cloud [GVB13, SZXW13]. AHP suit un processus en trois étapes :

1. *La modélisation de la structure hiérarchique :* dans cette étape le problème est défini et l'objectif est fixé. De la même façon, tous les critères ayant une influence dans la résolution du problème sont identifiés ainsi que les alternatives offrant une solution au problème. Dans la suite, les critères et les alternatives sont organisés dans une structure hiérarchique. Dans le cas d'un score unique exprimant la performance d'une infrastructure Cloud, nous avons créé une hiérarchie (Figure B.2) à partir des critères présentés ci-dessus (Section B.2.3) ; les alternatives correspondent aux différents fournisseurs de Cloud pris en charge par un Cloud broker. La performance de chaque alternative a été mesurée par un ensemble de logiciels de référence (Tableau B.1).

2. *Le classement de priorités :* des comparaisons par paires sont realisées pour déterminer l'importance relative de chaque critère et de chaque alternative. Pour cela, Saaty [Saa80] propose une échelle relative (Tableau B.2) sur laquelle l'évaluateur s'exprime sur l'importance relative d'un critère par rapport à un autre. Cette échelle permet de quantifier les comparaisons des paires de critères. Cette phase conduit à la construction de P matrices de comparaison de taille N *par* N, où N est le nombre d'alternatives et P est le nombre total de critères. Une matrice supplémentaire C

[2]Mega Instructions Per Second

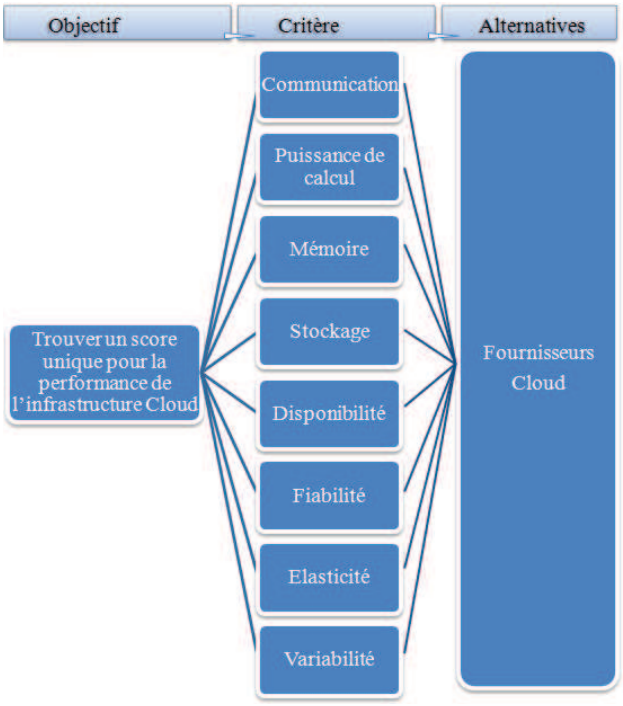

FIGURE B.2 – Hiérarchie du problème avec AHP

correspondant à la matrice de comparaison de critères est construite pour exprimer les poids relatifs entre chacun des critères à évaluer.

3. *La synthèse hiérarchique :* quand toutes les comparaisons ont été faites dans la deuxième étape, la probabilité numérique de chaque alternative est calculée. Cette probabilité détermine la vraisemblance que l'alternative atteigne l'objectif prévu. Cette procédure s'applique également à la matrice C qui exprime les poids relatifs entre chacun des critères. La phase de synthèse hiérarchique est appliquée aux matrices de comparaison de la façon suivante :

   (a) Synthèse de la matrice de comparaison par paires.

   (b) Calcul du vecteur de priorité.

   (c) Calcul de la valeur propre maximale.

   (d) Calcul de l'index de consistance.

   (e) Vérification de la consistance de la matrice de comparaison par paires.

   (f) Calcul des poids relatifs de chaque alternative ou critère.

Importance	Définition	Explication
1	Importance égale	Les deux critères contribuent également à l'objectif.
3	Faible importance d'un critère par rapport à l'autre	L'expérience et la comparaison favorisent légèrement un critère sur un autre.
5	Forte importance	L'expérience et la comparaison favorisent fortement un critère sur un autre.
7	Importance démontrée	Un critère est fortement favorisé et sa domination est démontrée dans la pratique.
9	Importance absolue	La preuve en faveur d'un critère par rapport à un autre est de l'ordre le plus élevé possible de l'affirmation.
2,4,6,8	Les valeurs intermédiaires entre deux comparaisons adjacentes	Quand le compromis est nécessaire entre deux critères.
Valeurs réciproques	Si le critère i a une des valeurs mentionnées ci-dessus lors de sa comparaison avec le critère j, alors j a la valeur réciproque lors de sa comparaison avec i	

TABLE B.2 – Échelle relative [Saa80]

B.3 Le placement dans les Clouds brokés

Le placement ou l'attribution des ressources dans un environnement de Cloud brokers correspond aux mécanismes pour distribuer l'infrastructure à travers de multiples Clouds basés sur les besoins et les contraintes des consommateurs. Les Cloud brokers peuvent réagir automatiquement aux scénarios imprévus dans lesquels les conditions de l'infrastructure Cloud changent, afin de maintenir l'opérabilité des applications des consommateurs. Voici deux exemples de scénarios où les Cloud brokers font appel aux algorithmes de placement :

- *Les changements des conditions du marché* : par exemple, l'introduction de nouvelles configurations de machines virtuelles, le changement de prix, l'apparition d'un nouveau fournisseur Cloud ou l'introduction d'une nouvelle politique de prix. Dans ce scénario, un Cloud broker déterminerait l'impact des changements des conditions du marché ou la performance des applications sur les gains des consommateurs d'infrastructure Cloud. Dans le cas d'un impact positif, l'utilisateur serait encouragé par le Cloud broker afin à migrer partiellement ou en totalité son infrastructure Cloud.
- *Les changements imprévus de l'infrastructure Cloud* : la panne d'un service Cloud peut gravement impacter les économies des consommateurs d'infrastructure. Même si dans la plupart de cas, les fournisseurs de Cloud offrent des compensations économiques aux consommateurs ayant subi la panne d'un service Cloud, ces compensations sont négligeables par rapport au fait d'avoir un service Cloud indisponible

(par exemple un site d'*e-commerce*). Dans ces scénarios, les Cloud brokers peuvent non seulement redéployer une infrastructure Cloud soumis à une panne, mais aussi minimiser le coût et le temps d'indisponibilité d'une application.

L'optimisation du placement consiste à choisir un ou plusieurs fournisseurs Cloud pour déployer un service en se basant sur des critères d'optimisation tels que le coût, la performance, etc. Les types de placement sont classés entre ceux basés sur des *exigences non-fonctionnelles* et ceux basés sur des *exigences de l'application*. Le placement basé sur des exigences non-fonctionnelles consiste à attribuer de l'infrastructure Cloud sur la base de paramètres tels que le nombre de processeurs et la quantité de mémoire et de stockage. D'autre part, le placement basé sur des exigences de l'application cherche à garantir la qualité de service en prenant en compte des paramètres spécifiques aux applications.

B.3.1 Placement basé sur des exigences non-fonctionnelles

Le placement basé sur des exigences non-fonctionnelles peut se faie dans des scénarios statiques ou dynamiques. Les placement suivant les scénarios statiques considèrent que les changements dans l'infrastructure n'arrivent jamais. A contrario, le placement suivant le scénario dynamique vis à reconfigurer d'une façon optimale l'infrastructure Cloud dans des nouvelles situations ou lors du changements des conditions. Les approches présentées dans la suite abordent le placement suivant des scénarios statiques et dynamiques à travers des modèles exacts.

Placement statique

Tordsson *et al.* [TMMVL12] proposent une architecture pour le Cloud brokering et un algorithme de placement basé sur la performance du logiciel GridNPB/ED et le prix des ressources d'infrastructure. Dans cette approche, les consommateurs ont la possibilité de contraindre le déploiement de l'infrastructure en spécifiant le nombre, le type et le pourcentage de machines virtuelles à déployer. Chaisiri *et al.* [CLN09] proposent un placement optimal de machines virtuelles à travers plusieurs Cloud providers en considérant l'approvisionnement en ressources par *réservation* à l'acte. L'approvisionnement par réservation implique un engagement de longue durée en échange d'une remise sur le prix de l'utilisation à la demande. Cependant, l'approvisionnement par réservation soulève également de nouvelles questions en cas de sous-dimensionnement ou de sur-dimensionnement des ressources en infrastructure. Dans le cas du sous-dimensionnement, le besoin en ressources d'infrastructure peut être totalement satisfait en achetant de la ressource en infrastructure au fur et à mesure avec un coût plus supérieur. Dans le cas de sur-dimensionnement, des questions sue l'entité (le consommateur ou le Cloud broker) à qui sont facturées les ressources d'infrastructure inutilisées.

L'utilité du placement de machines virtuelles pour les applications entièrement découplées ou faiblement couplées a été étudiée par Van den Bossche *et al.* [VdBVB10] et Moreno-Vozmediano *et al.* [MVML11]. Ces deux approches optimisent le coût du déploiement et considère la facturation à l'acte ainsi qu'une infrastructure Cloud hybride. D'une part, Van den Bossche *et al.* [VdBVB10] proposent un placement de coût optimal pour des charges de travail préemptibles mais non transférables parmi les fournisseurs, avec un délai strict d'exécution. Les charges de travail sont caractérisées par des exigences en mémoire, en puissance de calcul et en transmission de données. Les auteurs résolvent ce problème avec une programmation linéaire. D'autre part, Moreno-Vozmediano *et al.* [MVML11] évaluent le scénario du déploiement d'un cluster de calcul sur une infrastructure multi-Cloud pour résoudre des tâches faiblement couplées[3]. L'objectif de cette approche est d'optimiser le coût du déploiement ou de mettre en œuvre des stratégies de haute disponibilité. Cette approche est évaluée à l'aide d'un banc d'essai à petite échelle comprenant un centre de données local et trois différents Cloud publics. Les résultats obtenus à partir de ce banc d'essai ont été complétés par des simulations qui incluent un plus grand nombre de ressources.

Placement dynamique

Lucas-Simarro *et al.* [LSMVML11] proposent un algorithme de placement de machines virtuelles ayant pour but de minimiser le coût pour les consommateurs dans un environnement de prix dynamiques. Dans cette approche, le Cloud broker transfère l'infrastructure des consommateurs d'un fournisseur Cloud à un autre moins cher en fonction de variations de prix. L'algorithme calcule les nouveaux prix en fonction de la moyenne et de la tendance d'évolution du prix du fournisseur ; afin de garantir la performance des applications déployées sur les ressources d'infrastructure, les décisions de placement sont limitées par le nombre maximum et minimum de machines virtuelles par placement réattribuer et par l'exigence d'équilibrage de charge qui indique le pourcentage de ressources maximum à instancier au sein de chaque fournisseur de Cloud. Dans cette approche, le problème de placement est limité à une configuration de machine virtuelle. Lucas-Simarro [LSMVML12] étend ce travail en étudiant l'effet de plusieurs configurations de machines virtuelles et en abordant le problème d'optimisation de la performance. L'optimisation de la performance consiste à maximiser la performance des ressources déployées, en choisissant les machines virtuelles ayant la meilleure performance en termes de ressources matérielles (disque dur, mémoire, processeurs). Le principal inconvénient de cette approche est que les mesures de performance des machines virtuelles doivent être fournies par les consommateurs après avoir testé toutes les configurations des machines virtuelles au sein de chaque fournisseur de Cloud.

Un modèle plus complet aui vise non seulement l'optimisation des coûts mais aussi les

[3] *Loosely-coupled Many-Task Computing (MTC) applications*

changements imprévus de l'infrastructure Cloud par la migration de machines virtuelles a été proposé par Tordsson *et al.* [LTE11]. Dans ce modèle, le temps de migration d'une machine virtuelle est éstimé grâce au temps nécessaire pour arrêter une machine virtuelle chez un fournisseur de Cloud en plus du temps de redémarrage une nouvelle machine virtuelle (de configuration similaire à celle arrêtée) chez un autre fournisseur de Cloud.

Chaisiri *et al.* [CLN12] proposent un algorithme de minimisation de coûts pour l'approvisionnement de ressources d'infrastructure pour une certaine période étant donné l'incertitude de la demande et le prix. La décision optimale calculée par cet algorithme est basée sur la demande des consommateurs et le prix des fournisseurs de Cloud. Cela permet à un Cloud broker d'ajuster la quantité de ressources acquises à l'avance sous réservation et la quantité de ressources acquises à la demande, en tenant compte du fait que les machines virtuelles réservées à l'avance sont généralement moins chères que celles acquises à la demande. Cette approche aborde le problème du sous-dimensionnement et du sur-dimensionnement. Les auteurs résolvent ce problème à travers la programmation stochastique de nombres entiers.

B.3.2 Placement basé sur des exigences de l'application

Le placement basé sur des exigences de l'application fair varier dynamiquement les ressources d'infrastructure à travers plusieurs fournisseurs de Cloud sur la base des contraintes de qualité de service spécifiques à l'application. Dans le cas des applications étroitement couplées avec des exigences en communication, le processus de placement doit garantir un déploiement sur un seul fournisseur Cloud [GB12]. D'autre part, dans le cas des applications entièrement découplées [4] ou des applications faiblement couplées, le processus de placement peut profiter de l'hétérogénéité des offres de fournisseurs de Cloud pour fournir une solution rentable qui garantit une bonne performance de l'application [RCL09, VdBVB10]. Dans le cas des applications interactives (par exemple des jeux en ligne), l'expérience de l'utilisateur repose sur la bande passante et la latence causée par les distances géographiques [GB12]. Par conséquent, ce type d'applications pourrait être traité près de l'emplacement géographique d'origine pour obtenir une latence plus faible et un débit plus élevé.

L'importance des services de Cloud brokering pour les télécommunications est mise en évidence par Carella G. *et al.* [CMCS12]. Dans cette approche, un Cloud broker améliore ses mécanismes de placement sur la base des données en temps réel sur les performances du réseau, des exigences de la qualité de service et des prix des fournisseurs de Cloud. L'objectif est de fournir aux opérateurs de services de télécommunications une qualité de service minimale pour satisfaire les exigences des clients à l'aide de la surveillance

[4] Les applications sont entièrement découplées lorsque ses tâches n'ont aucune contraintes de précédence, elles peuvent donc être exécutées en parallèle.

des services déployés. Cette approche est évaluée à l'aide d'un banc d'essai composé d'un Cloud broker et d'un système IMS [5]. Le placement à coût optimal des applications Web 2.0 avec des exigences de haute disponibilité et de tolérance de panne à travers des fournisseurs de Cloud multiples a été proposé par Frincu *et al.* [FC11]. Dans cette approche, les auteurs considèrent des applications constituées par plusieurs composants et connecteurs (C/Cs). Les C/Cs sont réaffectés en faisant un enregistrement de l'état du C/C et en arrêtant la machine virtuelle qui les héberge. Après cela, une nouvelle machine virtuelle est démarrée, l'enregistrement est transmis et le C/C est redémarré dans l'état dans lequel il a été enregistré. Une architecture basée sur un Cloud broker intelligent qui réagit aux changements de processus opérationnels, en changeant la configuration de l'infrastructure au sein de plusieurs fournisseurs de Cloud est décrite par Grivas *et al.* [GKW10].

Le placement de services de qualités différentes et avec différentes exigences de provisionnement pour les applications d'*e-learning* a été abordé par Quarati *et al.* [QCGD13]. L'objectif de cette approche est de maximiser la satisfaction de l'utilisateur ainsi que les revenus du Cloud broker tout en réduisant les coûts d'énergie par le biais des mécanismes d'économie d'énergie. Pour cela, le Cloud broker attribue des ressources au sein de fournisseurs publics ou privés sur la base des attentes du consommateur en termes de qualité de service et également sur la base de la charge de travail de l'infrastructure Cloud privée. Cette approche a été évaluée à l'aide d'un simulateur à événements discrets.

B.3.3 Approche exacte au problème de placement en Cloud brokering

Dans cette thèse, le problème du placement de machines virtuelles en Cloud brokering a été modélisé comme un *problème du sac à dos* : étant donné un ensemble de machines virtuelles, avec chacune une configuration, un prix et une performance, il s'agit de déterminer le nombre de machines virtuelles de chaque configuration à fournir pour que l'infrastructure provisionnée soit égale ou supérieure à la requête du consommateur (c'est-à-dire la requête soit satisfaite) et pour que le coût de l'infrastructure Cloud soit aussi bas que possible (dans le cas d'une optimisation des coûts). Le problème de placement Cloud a été formulé comme un problème d'optimisation linéaire et la technique du *Goal programming* a été employée pour résoudre ce problème.

Dans cette thèse, une approche exacte du placement des ressources d'infrastructure Cloud à travers multiples fournisseurs est proposé ; elle peut être appliquée aux scénarios d'optimisation de coûts ainsi qu'aux scénarios de reprise après sinistre. Parmi les paramètres les plus importants, celle-ci prend en compte le prix, le type de machine virtuelle, les délais intrinsèques au réseau et la disponibilité du fournisseur de Cloud. L'originalité de notre approche réside dans l'association des configurations de machines virtuelles avec

[5] IP Multimedia Subsystem (IMS)

leurs performances respectives. La formulation du problème de placement est faite dans la section 4.3.

B.4 Les politiques de prix et les réservations faites à l'avance

B.4.1 Les politiques de prix en Cloud computing

Plusieurs modèles économiques issus d'autres domaines d'étude ont été proposés pour le Grid Computing [BAGS02a]. Les modèles du marché des produits de base, de prix affichés, d'appel d'offres, de négociation et de vente aux enchères sont parmi les modèles économiques les plus couramment étudiés pour la gestion des ressources dans le Cloud [BAGS02b]. Cependant, la plupart d'entre eux n'ont pas été mis en pratique par les fournisseurs actuels de Cloud. Le « *Pay-as-you-go* » (facturation à l'acte) et les politiques *par abonnement* sont parmi les politiques de prix les plus populaires appliquées par les fournisseurs de Cloud actuels [WABS09]. Dans le modèle pay-as-you-go, les utilisateurs paient un montant proportionnel à leur consommation de ressources tandis que dans les politiques par abonnement les consommateurs doivent s'engager à utiliser le service pendant une période de temps donnée, en échange de quoi ils paient un prix plus bas par unité de temps que dans le pay-as-you-go. Généralement, les ressources achetées à travers des politiques d'abonnement ont priorité en termes de disponibilité par rapport à celles où les ressources sont acquises par pay-as-you-go. Parmi les politiques de prix de Cloud déployées par les fournisseurs actuels, on trouve :

1. *Freemium :* un produit ou un service est gratuit, mais les utilisateurs doivent payer pour les fonctionnalités avancées. L'usage du produit ou service peut être limité en temps, en capacité, en qualité du service, en caractéristiques, etc. (par exemple *les machines virtuelles du niveau d'utilisation gratuit d'Amazon EC2*).

2. *Facturaion à l'acte ou pay-as-you-go :* les utilisateurs paient un montant proportionnel à leur consommation réelle de ressources (par ex. *les machines virtuelles à la demande d'Amazon EC2*).

3. *Par abonnement :* les utilisateurs s'engagent à utiliser le service pendant une période de temps donné, en échange de quoi ils paient un prix réduit sur le long terme par rapport au pay-as-you-go. Cette politique permet aux fournisseurs d'anticiper l'utilisation de leur infrastructure Cloud et d'accélérer leur retour sur investissement. Dans cette politique de prix, l'attribution des ressources est basée sur les réservations faites à l'avance (RFA). Grâce aux RFA, les fournisseurs de Cloud bloquent des ressources et garantissent leurs disponibilités futures aux consommateurs [LRY+11]. La politique de prix par abonnement peut être divisée en trois catégories :

- Forfaitaire : les utilisateurs sont facturés à un prix fixe pour une période de temps donnée indépendamment de l'utilisation des ressources (par exemple *les machines virtuelles réservées à une utilisation intensive d'Amazon EC2*) .

- Abonnement avec un quota : les utilisateurs doivent payer des frais fixes pour s'abonner au service et couvrir un quota d'utilisation. Si le quota est épuisé, il y a des frais pour la consommation supplémentaire.

- Abonnement sans quota : les utilisateurs sont facturés un montant fixe pour souscrire au service avec un supplément en fonction de l'utilisation (par exemple *machines virtuelles réservées pour une utilisation légère ou modérée d'Amazon EC2*).

4. *Sur la base du marché :* les utilisateurs font des enchères pour acquérir les ressources, les ressources sont allouées si l'enchère dépasse le prix fixé par le fournisseur de Cloud (par exemple *les instances ponctuelles d'Amazon EC2*). Les fournisseurs utilisent cette politique de prix pour vendre leurs capacités inutilisées d'infrastructure Cloud.

Les utilisateurs sélectionnent une politique de prix en fonction de leurs besoins tels que la puissance de calcul, la mémoire, le stockage, la qualité de service, le temps d'exécution, le budget, etc. Ainsi, les consommateurs contraints par le temps dans l'exñecution de leurs tâches seraient plus intéressés par l'achat d'un abonnement de machines virtuelles, afin d'assurer leur disponibilité à tout moment. Au contraire, les utilisateurs qui souhaite exécuter des tâches sans contraintes de temps seraient prêts à louer des machines virtuelles via la politique de prix basée sur le marché. Dans le cas des scénarios à tâches variables et imprévisibles, les machines virtuelles sont louées via la politique pay-as-you-go.

B.4.2 Les réservations faites à l'avance

Les réservations faites à l'avance (RFA) ont été introduites pour garantir de manière efficace la disponibilité d'une quantité de ressources donnée à utiliser à un moment déterminé dans le future. La réservation de chambres d'hôtel est un des meilleurs exemples de RFA. Dans ce cadre, une RFA est décrite par au moins trois paramètres : le nombre de chambres à réserver, les dates d'arrivée et du départ. Les mécanismes pour gérer les RFA ont été appliqués à plusieurs problèmes de partage des ressources en informatique tels que la réservation de bande passante, la planification des tâches et la planification de machines virtuelles. Dans ce qui suit est présentée une classification de certaines études portant sur les RFA en informatique.

Les réservations faites à l'avance par les fournisseurs de Cloud

Ce type de RFA est étroitement lié à la politique de prix par abonnement, largement proposée par les fournisseurs de Cloud. Ce type de réservation fonctionne sur une base d'intervalle de temps. Au début de chaque intervalle de temps, l'utilisateur peut ajuster la quantité de ressources à réserver par le fournisseur de Cloud pour le prochain intervalle de temps. Les études menées sur ce type de réservation peuvent être classées en deux catçegories : les plans de réservation à court terme [NFL12b, NFL12a] (par exemple avec des intervalles de temps de 10 minutes ou 1 heure) et les plans de réservation à long terme (par exemple des intervalles de temps de plusieurs années) [SAMVML11, CLN12].

Niu, D. *et al.* [NFL12b] ont étudié les politiques de prix pour la réservation de bande passante dans le Cloud sur un plan à court terme de l'ordre de quelques heures ou quelques dizaines de minutes. Les requêtes des consommateurs sont caractérisées par une estimation de la moyenne de la bande passante, de leur variabilité et le pourcentage du trafic à être garantit par la bande passante demandée. Dans cette approche, le fournisseur de Cloud calcule la réservation de bande passante courante d'une façon probabiliste afin de garantir les performances requises. Il décide également des frais de réservation en tenant compte des rafales de rêquetes et de la corrélation temporelle des différentes requêtes. Un problème similaire dans lequel un broker est introduit entre les fournisseurs de Cloud et les utilisateurs est également étudié par Niu, D. *et al.* [NFL12a]. Dans cette approche, un broker vend individuellement des garanties de bande passante pour les utilisateurs. Pour cela, le broker réserve conjointement la bande passante chez plusieurs fournisseurs de Cloud Computing et il exploite le multiplexage statistique pour réduire les coûts de réservation. Ce problème a été résolu en utilisant la théorie des jeux où le prix d'équilibre de la bande passante dépend de l'attente lié au nombre de requêtes, des rafales de requêtes ainsi que de sa corrélation avec le marché.

Le plan de réservation à long terme a été étudié d'abord par San-Aniceto, I. *et al.* [SAMVML11]. Dans cette approche est considéré et un seul algorithme est proposé pour sélectionner le nombre de machines virtuelles qui doivent être réservées par un utilisateur tout en déployant un service dans le Cloud. Afin de faire face aux fluctuations et au caractère imprévisible de la demande, des ressources supplémentaires peuvent être provisionnées dynamiquement grâce au modèle on-demand. L'algorithme proposé réduit le coût global des machines virtuelles acquises en tirant parti des diffèrents politiques de prix au sein d'un Cloud provider. Chaisiri, S. *et al.* [CLN12] ont généralisé le problème décrit précédemment au contexte de plusieurs fournisseurs de Cloud en tenant compte de l'incertitude des requêtes des utilisateurs et du prix futur des ressources fixé par les fournisseurs. Ce problème a été modélisé sous la forme d'un programme stochastique entier et a été résolu numériquement.

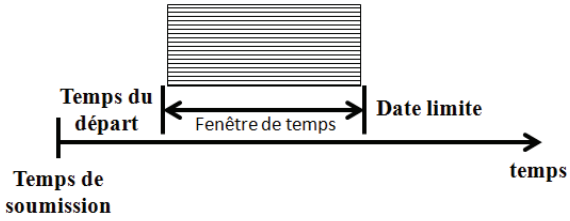

FIGURE B.3 – RFA avec un temps de démarrage et une date limite stricts

Les réservations faites à l'avance par les consommateurs

Dans ce type de RFA, les utilisateurs disposent d'une plus grande flexibilité car ils peuvent indiquer non seulement leurs exigences en matière de capacité mais aussi diverses contraintes de temps liées à l'exécution de leurs tâches. Les contraintes de temps peuvent être exprimées en fonction de divers paramètres tels que le temps de dépôt de la requête, le temps de démarrage, le temps d'éxécution (ou durée) ou la date limite pour accomplir une tâche. Ainsi, les utilisateurs ont la possibilité de réserver à l'avance les ressources estimées comme nécessaires pour l'accomplissement de leurs tâches sans aucun engagement. Nous définissons *la fenêtre de temps d'une RFA* comme l'intervalle délimité par le temps de démarrage et la date limite pour accomplir une tâche d'une RFA. Les RFA spécifiées par les utilisateurs peuvent être classées en trois grandes catégories :

1. ***RFA avec un temps de démarrage et une date limite stricts :*** ce type de RFA est caractérisé par une durée égale à sa fenêtre temporelle. Les utilisateurs ont besoin de ressources dans le futur à un instant et pour une durée bien précis (figure B.3). Cette RFA, ne laisse aucune flexibilité au fournisseur de Cloud de reporter la tâche à une période différente. Plusieurs études ont montré que les RFA avec un temps de démarrage et une date limite stricts conduisent à une grande fragmentation de la disponibilité des ressources en augmentant le nombre d'intervalles de temps inutilisés [SFT00, THW02]. Dans le cas des machines virtuelles en Cloud computing, ces intervalles de temps peuvent être utilisés par d'autres types de requête telles que celles pour les machines virtuelles on-demand ou les machines virtuelles vendues au travers d'une place du marché.

   Aoun, R. *et al.* [ADG10] ont effectué des travaux de recherche sur l'approvisionnement de ressources informatiques pour répondre au mieux aux demandes de RFA. Les auteurs ont considéré plusieurs services de base et ont mis en évidence la distribution du stockage de données et le transfert des données via la multidiffusion afin de satisfaire un plus grand nombre d'utilisateurs et améliorer l'utilisation des ressources chez les fournisseurs de Cloud. Dans d'autres études, les mêmes auteurs ont fait un plan d'affaires du problème mentionné ci-dessus [AG09c]. Le plan proposé compare trois politiques de prix en fonction des attentes des utilisateurs et des fournisseurs de Cloud.

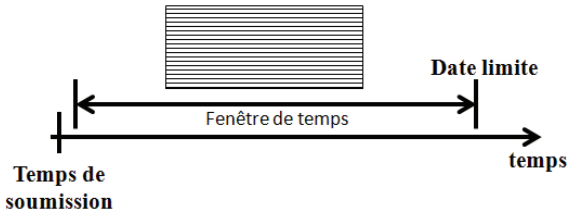

FIGURE B.4 – RFA avec un temps de démarrage flexible et une date limite stricte

2. ***RFA avec un temps de démarrage flexible et une date limite stricte :*** ce type de RFA est caractérisé par une flexibilité plus élevée que la RFA présentée ci-dessus car elle considère une fenêtre de temps plus large ; lorsque ces RFA sont acceptées le Cloud provider doit garantir les exécutions de tâches avant la date limite (figure B.4). Les fournisseurs de Cloud peuvent utiliser des mécanismes divers pour organiser, gérer et contrôler efficacement leurs ressources. Par exemple, Lu, K. *et al.* [LRY+11] introduisent un modèle basé sur la géométrie algorithmique qui permet aux fournisseurs d'enregistrer et de vérifier la disponibilité de leurs ressources au cours de la phase de négociation et de planification du contrat (Service Level Agreement (SLA), en anglais). Dans ce modèle, lorsque le fournisseur manque de ressources, une ou plusieurs solutions alternatives flexibles appelées contre-offres, peuvent être générées afin de satisfaire l'utilisateur. Par conséquent, la réputation du fournisseur est renforcée par sa capacité à satisfaire le plus grand nombre possible d'utilisateurs, afin de mieux utiliser les ressources et d'obtenir des profits plus élevés. Venugopal, S. *et al.* [VCB08] proposent un mécanisme de négociation qui permet de modifier le SLA ou de faire des contre-propositions aux deux parties (fournisseurs de Cloud et utilisateurs) afin d'aboutir à un accord. Dans les scénarios étudiés, une fois que le SLA a été convenu, le fournisseur doit exécuter la tâche à l'heure spécifiée. Des simulations numériques ont été réalisées pour mettre en évidence le bénéfice apporté par les RFA de ce type. Kaushik, N. *et al.* [KFC06] a étudié l'impact de la taille de la fenêtre sur la probabilité de blocage et l'utilisation des ressources pour divers modèles d'inter-arrivées et de temps de service dans le cadre de la politique d'ordonnancement FIFO.

   Aoun, R. *et al.* [AG09a] ont étudié le problème de l'approvisionnement des ressources dans un marché Cloud en prenant en compte les RFA avec une fenêtre flexible dans le temps, étant donnée la taille de la fenêtre en fonction des exigences et des budgets des utilisateurs. Le but de cette étude est de proposer un algorithme de gestion équitable qui garantit la qualité de service et les exigences des utilisateurs tout en augmentant le bénéfice attendu des fournisseurs. A cette fin, les auteurs ont introduit une fonction de coût pondérée qui permet la différenciation des services en s'appuyant sur les disparités de durée des RFA. Une formulation linéaire exacte [AG09a] ainsi qu'une approche heuristique [AG09b] ont été prises

en compte pour l'évaluation numérique de la performance de cette approche. Au lieu de payer des prix fixes, Yeo, C.S. *et al.* [YVCB10] proposent d'ajuster automatiquement le prix pour l'accès aux ressources si nécessaire afin d'augmenter les revenus des fournisseurs de Cloud. En utilisant des prix variables, les fournisseurs peuvent inciter les utilisateurs décaler l'utilisation du service aux périodes creuses en bénéficiant de prix plus bas. Comme les prix sont ajustés en fonction de la charge de travail prévue et de la disponibilité des ressources, les RFA soumises longtemps à l'avance sont privilégiées avec des prix inférieurs par rapport aux RFA tardives.

Un environnement permettant aux fournisseurs de Cloud de modifier le calendrier d'exécution des RFA déjà acceptées afin de lancer des nouvelles RFA a été proposé par Netto, M. *et al.* [NBB07]. Ce rééchelonnement des RFA est effectué en respectant les contraintes de temps d'exécution indiquées dans le SLA. Les auteurs ont montré que ce mécanisme peut atténuer les effets négatifs des RFA et améliorer la performance des ordonnanceurs. Il permet en effet de réduire les intervalles de temps où les ressources restent inutilisées. Une autre solution pour améliorer l'utilisation des ressources est de faire usage des mécanismes de surréservation qui sont particulièrement efficaces dans les scénarios avec des politiques d'annulation des RFA [SKB08] et de surestimation du temps d'exécution des RFA [BB11].

Dans ce contexte, le rééchelonnement des RFA existantes peut permettre aux RFA sur-réservées d'avoir accès aux ressources pour leur période d'exécution si les RFA précédentes sont annulées par l'utilisateur ou terminent plus tôt. L'algorithme d'ordonnancement préemptif à échéance proche a été étudié pour fournir des garanties probabilistes en temps réel pour les RFA sur des machines à temps partagé [KKV+09]. Avec cette stratégie de planification, une politique de contrôle d'admission est développée dans laquelle de nouvelles RFA sont acceptées si elles ne violent pas les contraintes de qualité de service de réservations préalablement acceptées. Ceci peut être réalisé, par exemple, en changeant la priorité des RFA afin de faire en sorte que leur exécution se termine avant leur date limite.

3. ***RFA avec un temps de démarrage et une date limite flexibles :*** ce type de RFA est également caractérisé par une flexibilité élevée. Cependant, la fenêtre n'est pas clairement définie. Au lieu de définir un temps de démarrage et une date limite pour l'exécution de chaque RFA, l'utilisateur fournit un ensemble d'intervalles de temps selon ses préférences représenté par une fonction d'utilité (figure B.5). La fonction d'utilité représente le niveau de satisfaction que l'utilisateur final obtient à la suite de la négociation. Cette satisfaction peut dépendre de plusieurs paramètres tels que le temps d'exécution, le prix des ressources ou encore les exigences de qualité de service, etc. En général, le résultat le plus défavorable est celui où l'utilisateur et le fournisseur ne sont pas en mesure de parvenir à un accord ; dans ce cas l'utilisateur reçoit une utilité nulle car sa demande est rejetée. Des politiques

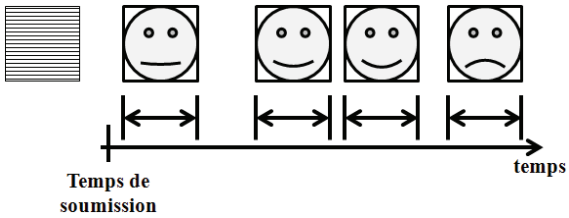

FIGURE B.5 – RFA avec un temps de départ et une date limite flexibles. L'humeur de l'émoticône représente la satisfaction d'un utilisateur pour un intervalle de temps.

de prix dynamiques basées sur l'utilisation des ressources et sur le classement des utilisateurs ont été étudiées par Püschel, T. *et al.* [PN09]. De telles stratégies de prix dynamiques permettent d'adapter le prix afin d'encourager l'utilisation des ressources pendant les périodes creuses. Deux approches différentes, déjà bien établies dans d'autres domaines, sont comparées par Meinl, T. *et al.* [MAT10], à savoir, la réservation faite en situation de concurrence parfaite entre fournisseurs de Cloud et par la gestion du rendement en dans un environnement de concurrence imparfaite. Les auteurs analysent les différentes exigences en vue d'appliquer les solutions proposées dans le Cloud et de fournir des modèles pour calculer les prix de réservations appropriés. Fils, S. *et al.* [SS12] introduisent un mécanisme de négociation bilatérale pour la réservation des services de Cloud qui tient compte simultanément du prix et du temps d'exécution. Des simulations numériques ont été utilisées pour étudier le mécanisme proposé dans le cadre des politiques de prix, traditionnellement utilisées par les fournisseurs de Cloud, à savoir les politiques à prix fixe pour les machines virtuelles réservées et à la demande, et les politiques de prix variable pour les machines virtuelles ponctuelles (celles achetées sur une place de marché).De plus, une politique de prix se bassant sur l'heure d'utilisation des ressources a été étudiée par Saure, D. *et al.* [SSQ+10]. D'après cette politique, le prix des ressources est totalement indépendant du taux d'utilisation des ressources demandées mais varie au cours de la journée. Une stratégie de prix optimale maximisant la satisfaction de l'utilisateur a été conçue dans cette approche.

Dans le cas des RFA sous-estimées, les RFA se déroulent pendant une période plus longue que prévu. Yeo, CS *et al.* [YVCB10] traitent le problème de RFA sous-estimées avec *un temps de démarrage flexible et avec une date limite stricte*. Toutefois, afin de respecter les RFA futures, les RFA en cours d'exécution sont arrêtés une fois la période de réservation dépassée.

B.4.3 La politique de prix pay-as-you-book

Pay-as-you-book est une politique de prix hybride combinant les avantages des politiques de *pay-as-you-go* et celles *par abonnement*. Elle consiste à payer et à réserver à l'avance

Caractéristique	Pay-as-you-go	Par abonnement	Pay-as-you-book
Coût	Élevé	Bas	Moyen
Coûts de l'utilisateur	Variable	Variable et fixe	Fixe, sauf si présence de RFA est sous-estimées
Compensation en cas d'indisponibilité du service	Aucune	Pourcentage de frais d'utilisation	X fois le prix de la RFA
Modalités de paiement	À échéance	À l'avance	À terme échu ou à l'avance
Engagements à terme	Aucun	Long (De quelques mois à plusieurs années)	Court (Durée de la RFA)
Disponibilité pendant les périodes de très forte demande	Base	Haute	Dépend des politiques du fournisseur
Utilisation de l'analyse prédictive	Modèles d'utilisation imprévisibles	Nécessaire et fait par le fournisseur	Non nécessaire car la prédiction est effectuée par l'utilisateur
Type d'application	Avec un profil d'utilisation imprévisible	Avec un profil d'utilisation prévisible	Avec un profil d'utilisation très prévisible

TABLE B.3 – Comparaison des politiques de prix les plus utilisées avec pay-as-you-book

des créneaux horaires pour l'utilisation de machines virtuelles. Le prix de ces créneaux n'implique pas de frais d'abonnement ou d'engagement à long terme. De cette façon, pay-as-you-book permet d'éviter la dépendance exclusive à l'égard d'un fournisseur introduite par les politiques par abonnement. Un autre avantage de pay-as-you-book est un coût d'usage fixe car les utilisateurs paient pour ce qu'ils ont réservé. Cela représente également un avantage pour les fournisseurs de Cloud qui pourraient considérablement réduire ou éviter l'utilisation de techniques d'analyse prédictive afin de déterminer les tendances d'utilisation. Le Tableau B.3 présente une comparaison des politiques de prix les plus employées actuellement par les fournisseurs de Cloud et de pay-as-you-book.

Le pay-as-you-book peut être appliquée dans des cas d'usage avec des profils d'utilisation prévisibles [HY10] tels que :

- **le schéma récurrent dans une journée :** scénarios avec des cycles récurrents dans la consommation de ressources basée sur les profils d'utilisation personnels, par exemple la consommation des ressources informatiques par les utilisateurs d'une entreprise peut être facilement prédite et décrite comme le besoin de R ressources entre 8h et 17h du lundi au vendredi, où R est calculé en fonction du nombre d'utilisateurs et de la quantité de ressources utilisée par utilisateur ;

- **la variabilité propre à une industrie :** scénarios avec une variabilité prévisible en fonction des événements récurrents, comme le période des impôts, la Coupe du Monde de foot, pèriode de Noël, etc.

B.5 Conclusions et travaux futurs

L'objectif de cette thèse a été de proposer des nouveaux services à valeur ajoutée et une politique de prix dans le Cloud brokering au niveau de l'infrastructure. L'application pratique de cette thèse est triple. Tout d'abord, le facteur de qualité de la performance proposé peut être utilisé pour comparer objectivement les fournisseurs de Cloud sur la base de leur performance et du profil d'application à déployer. D'autre part, le facteur de performance lié à l'algorithme de placement proposé apporte une allocation des ressources à coût optimal en fonction des contraintes de l'utilisateur. Ainsi, cet algorithme peut enrichir l'offre de services non seulement des Cloud brokers, qui pourraient réagir automatiquement à des scénarios imprévus, mais aussi des entreprises de conseil et de services informatiques qui peuvent prendre des décisions lors de la migration des applications vers le Cloud. Enfin, la politique de prix proposée représente une première étape pour l'étude des nouveaux moyens attractifs pour acheter de l'infrastructure Cloud.

Ce travail a permis d'identifier deux axes possibles d'approfondissement. Il s'agit notamment de l'identification de configurations types de machines virtuelles et la mise en place de SLA standards. L'identification de configurations types de machines virtuelles résout le problème de la disparité de l'offre de machines virtuelles actuellement présente chez les fournisseurs de Cloud. Ainsi, le défi est d'identifier une jauge de configurations de machines virtuelles qui satisfasse au plus grande nombre de demande des utilisateurs et qui prend en compte les différents profils d'application. D'autre part, les SLAs varient actuellement parmi les fournisseurs de Cloud. Afin de permettre la comparaison des offres de service, les attributs des SLA et leurs définitions doivent être normalisés entre les fournisseurs de Cloud. En résumé, les SLA standards, les configurations de machines virtuelles types ainsi que le facteur de performance proposé dans cette thèse contribuent faire des machines virtuelles un bien d'utilité publique.

Bibliography

[ADG10] R. Aoun, E.A. Doumith, and M. Gagnaire. Resource Provisioning for Enriched Services in Cloud Environment. In *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 296–303, Nov.-Dec. 2010.

[ADWC10] Mohammed Alhamad, Tharam Dillon, Chen Wu, and Elizabeth Chang. Response time for cloud computing providers. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '10, pages 603–606, New York, NY, USA, 2010. ACM.

[AG09a] R. Aoun and M. Gagnaire. An Exact Optimization Tool for Market-Oriented Grid Middleware. In *IEEE International Workshop on Communications Quality and Reliability (CQR)*, pages 1–4, May 2009.

[AG09b] R. Aoun and M. Gagnaire. Service Differentiation Based on Flexible Time Constraints in Market-Oriented Grids. In *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–8, Nov.-Dec. 2009.

[AG09c] R. Aoun and M. Gagnaire. Towards a Fairer Benefit Distribution in Grid Environments. In *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 21–26, May 2009.

[aws] Amazon elastic compute cloud: Micro instances (t1.micro). http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts_micro_instances.html, Accessed: June 2014,.

[BAGS02a] Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger. Economic models for resource management and scheduling in grid computing. In *The Journal of Concurrency and Computation: Practice and Experience (CCPE)*, pages 1507–1542. Wiley Press, 2002.

[BAGS02b] Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger. Economic Models for Resource Management and Scheduling in Grid Computing. *Concurrency and Computation: Practice and Experience*, 7(13-15):1507–1542, 2002.

[BB11] G. Birkenheuer and A. Brinkmann. Reservation-Based Overbooking for HPC Clusters. In *IEEE International Conference on Cluster Computing (CLUSTER)*, pages 537–541, Sep. 2011.

[BK10] Christian Baun and Marcel Kunze. Performance measurement of a private cloud in the opencirrus™ testbed. In *Proceedings of the 2009 International Conference on Parallel Processing*, Euro-Par'09, pages 434–443, Berlin, Heidelberg, 2010. Springer-Verlag.

[Cac] Cachebench: benchmark to evaluate the performance of the memory hierarchy of computer systems. http://icl.cs.utk.edu/projects/llcbench/cachebench.html, Accessed: June 2014,.

[Can12] M. Cantara. Gartner: Hype cycle for cloud services brokerage, July 2012.

[CLN09] S. Chaisiri, Bu-Sung Lee, and D. Niyato. Optimal virtual machine placement across multiple cloud providers. In *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, pages 103–110, 2009.

[CLN12] S. Chaisiri, Bu-Sung Lee, and D. Niyato. Optimization of resource provisioning cost in cloud computing. *Services Computing, IEEE Transactions on*, 5(2):164–177, 2012.

[CMCS12] Giuseppe Carella, Thomas Magedanz, Konrad Campowsky, and Florian Schreiner. Network-aware cloud brokerage for telecommunication services. In *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, pages 131–136, 2012.

[COn] Compatibleone: The opensource cloud broker. http://www.systematic-paris-region.org/fr/projets/compatible-one, Accessed: June 2014.

[cra] c-ray: benchmark for measuring floating point cpu performance. http://www.futuretech.blinkenlights.nl/c-ray.html, Accessed: June 2014,.

[DPC09] Jiang Dejun, Guillaume Pierre, and Chi-Hung Chi. Ec2 performance analysis for resource provisioning of service-oriented applications. In *Proceedings of the 2009 International Conference on Service-oriented Computing*, ICSOC/ServiceWave'09, pages 197–207, Berlin, Heidelberg, 2009. Springer-Verlag.

[DTM11] W. Dawoud, I. Takouna, and C. Meinel. Elastic vm for rapid and optimum virtualized resources' allocation. In *Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on*, pages 1–4, Oct 2011.

[EKKJP10] Yaakoub El-Khamra, Hyunjoo Kim, Shantenu Jha, and Manish Parashar. Exploring the performance fluctuations of hpc workloads on clouds. In *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, CLOUDCOM '10, pages 383–387, Washington, DC, USA, 2010. IEEE Computer Society.

[FC11] M.E. Frincu and C. Craciun. Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, pages 267–274, 2011.

[Fel13] W. Fellows. Cloud brokers: Now seeking ready-to-pay customers, January 2013.

[GB12] Nikolay Grozev and Rajkumar Buyya. Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, pages n/a–n/a, 2012.

[GKW10] S.G. Grivas, T.U. Kumar, and H. Wache. Cloud broker: Bringing intelligence into the cloud. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 544–545, 2010.

[GVB13] Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. A framework for ranking of cloud computing services. *Future Gener. Comput. Syst.*, 29(4):1012–1023, June 2013.

[HLM+10] Zach Hill, Jie Li, Ming Mao, Arkaitz Ruiz-Alvarez, and Marty Humphrey. Early observations on the performance of windows azure. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 367–376, New York, NY, USA, 2010. ACM.

[HY10] Rolf Harms and Michael Yamartino. The economics of the cloud. Microsoft whitepaper, Microsoft Corporation, Redmond, WA, USA, November 2010.

[HZKD11] Q. He, S. Zhou, B. Kobler, and T. Duffy, D. ad McGlynn. Performance analysis of cloud computing services for many-tasks scientific computing. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):931–945, 2011.

[ILFL12] Sadeka Islam, Kevin Lee, Alan Fekete, and Anna Liu. How a consumer can measure elasticity for cloud platforms. In *Proceedings of the 3rd*

ACM/SPEC International Conference on Performance Engineering, ICPE '12, pages 85–96, New York, NY, USA, 2012. ACM.

[IOY+11a] A. Iosup, S. Ostermann, M.N. Yigitbasi, R. Prodan, T. Fahringer, and D. H J Epema. Performance analysis of cloud computing services for many-tasks scientific computing. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):931–945, 2011.

[IOY+11b] A. Iosup, S. Ostermann, M.N. Yigitbasi, R. Prodan, T. Fahringer, and D. H J Epema. Performance analysis of cloud computing services for many-tasks scientific computing. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):931–945, 2011.

[ioz] Iozone: filesystem benchmark tool. http://www.iozone.org/, Accessed: June 2014,.

[IPE12] A. Iosup, R. Prodan, and D. Epema. Iaas cloud benchmarking: Approaches, challenges, and experience. In *proceedings of the ACM/IEEE Conference on High Performance Networking and Computing (SC), MTAGS*, pages 1–8. IEEE/ACM, 2012.

[KFC06] N.R. Kaushik, S.M. Figueira, and S.A. Chiappari. Flexible Time-Windows for Advance Reservation Scheduling. In *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 218–225, Sep. 2006.

[KKV+09] K. Konstanteli, D. Kyriazis, T. Varvarigou, T. Cucinotta, and G. Anastasi. Real-Time Guarantees in Flexible Advance Reservations. In *IEEE International Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 67–72, Jul. 2009.

[Kos91] Eric Kostlan. Statistical complexity of dominant eigenvector calculation. *Journal of Complexity*, 7(4):371 – 379, 1991.

[LML+11] A. Lenk, M. Menzel, J. Lipsky, S. Tai, and P. Offermann. What are you paying for? performance benchmarking for infrastructure-as-a-service offerings. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 484–491, 2011.

[LOCZ12] Zheng Li, Liam OBrien, Rainbow Cai, and He Zhang. Towards a taxonomy of performance evaluation of commercial cloud services. *2013 IEEE Sixth International Conference on Cloud Computing*, 0:344–351, 2012.

[LOZC12] Zheng Li, Liam O'Brien, He Zhang, and Rainbow Cai. On a catalogue of metrics for evaluating commercial cloud services. In *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*,

GRID '12, pages 164–173, Washington, DC, USA, 2012. IEEE Computer Society.

[LOZC13] Zheng Li, L. O'Brien, He Zhang, and R. Cai. Boosting metrics for cloud services evaluation – the last mile of using benchmark suites. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 381–388, 2013.

[LRY+11] Kuan Lu, T. Roblitz, R. Yahyapour, E. Yaqub, and C. Kotsokalis. QoS-aware SLA-based Advanced Reservation of Infrastructure as a Service. In *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 288–295, Nov.-Dec. 2011.

[LSMVML11] J.L. Lucas Simarro, R. Moreno-Vozmediano, R.S. Montero, and I.M. Llorente. Dynamic placement of virtual machines for cost optimization in multi-cloud environments. In *High Performance Computing and Simulation (HPCS), 2011 International Conference on*, pages 1–7, 2011.

[LSMVML12] J.L. Lucas-Simarro, R. Moreno-Vozmediano, R.S. Montero, and I.M. Llorente. Scheduling strategies for optimal service deployment across multiple clouds. *Future Generation Computer Systems*, in press, 2012.

[LTE11] Wubin Li, J. Tordsson, and E. Elmroth. Modeling for dynamic cloud scheduling via migration of virtual machines. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pages 163–171, 2011.

[LYKZ10] Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. Cloudcmp: comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 1–14, New York, NY, USA, 2010. ACM.

[LZO+13] Zheng Li, He Zhang, Liam O'brien, Rainbow Cai, and Shayne Flint. On evaluating commercial cloud services: A systematic review. *J. Syst. Softw.*, 86(9):2371–2393, September 2013.

[MAT10] T. Meinl, A. Anandasivam, and M. Tatsubori. Enabling Cloud Service Reservation with Derivatives and Yield Management. In *IEEE Conference on Commerce and Enterprise Computing (CEC)*, pages 150–155, Nov. 2010.

[MH12] M. Mao and M. Humphrey. A performance study on the vm startup time in the cloud. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 423–430, 2012.

[MVML11] Rafael Moreno-Vozmediano, Ruben S. Montero, and Ignacio M. Llorente. Multicloud deployment of computing clusters for loosely coupled mtc applications. *IEEE Trans. Parallel Distrib. Syst.*, 22(6):924–930, June 2011.

[NBB07] Marco A. Netto, Kris Bubendorfer, and Rajkumar Buyya. SLA-Based Advance Reservations with Flexible and Adaptive Time QoS Parameters. In *International Conference on Service-Oriented Computing (ICSOC)*, pages 119–131, 2007.

[NFL12a] Di Niu, Chen Feng, and Baochun Li. A Theory of Cloud Bandwidth Pricing for Video-on-Demand Providers. In *IEEE INFOCOM*, pages 711–719, Mar. 2012.

[NFL12b] Di Niu, Chen Feng, and Baochun Li. Pricing Cloud Bandwidth Reservations Under Demand Uncertainty. In *ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, pages 151–162, 2012.

[PEP11] S.C. Phillips, V. Engen, and J. Papay. Snow white clouds and the seven dwarfs. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pages 738–745, 2011.

[pho] Phoronix test suite: The leading software for automated, open-source testing & benchmarking. http://www.phoronix-test-suite.com/, Accessed: June 2014,.

[PN09] T. Püschel and D. Neumann. Management of Cloud Infrastructures: Policy-Based Revenue Optimization. In *International Conference on Information Systems (ICIS)*, Dec. 2009.

[QCGD13] Alfonso Quarati, Andrea Clematis, Antonella Galizia, and Daniele DâĂŹAgostino. Hybrid clouds brokering: Business opportunities, qos and energy-saving issues. *Simulation Modelling Practice and Theory*, 1(0):–, 2013.

[RCL09] B.P. Rimal, Eunmi Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, pages 44–51, 2009.

[Saa80] T.L. Saaty. *The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation.* McGraw-Hill, New york, 1980.

[Saa05] T.L. Saaty. *Theory and Applications of the Analytic Network Process. Pittsburgh.* RWS Publications, 4922 Ellsworth Avenue, Pittsburgh, PA 15213, 2005.

[SAMVML11] I. San-Aniceto, R. Moreno-Vozmediano, R.S. Montero, and I.M. Llorente. Cloud Capacity Reservation for Optimal Service Deployment. In *IARIA International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 52–59, Sep. 2011.

[SASA+11] K. Salah, M. Al-Saba, M. Akhdhor, O. Shaaban, and M. I. Buhari. Performance evaluation of popular cloud iaas providers. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 345–349, 2011.

[SDQR10] Jörg Schad, Jens Dittrich, and Jorge-Arnulfo Quiané-Ruiz. Runtime measurements in the cloud: Observing, analyzing, and reducing variance. *Proc. VLDB Endow.*, 3(1-2):460–471, September 2010.

[SFT00] W. Smith, I. Foster, and V. Taylor. Scheduling with Advanced Reservations. In *International Parallel and Distributed Processing Symposium (IPDPS)*, pages 127–132, 2000.

[SKB08] A. Sulistio, Kyong Hoon Kim, and R. Buyya. Managing Cancellations and No-Shows of Reservations with Overbooking to Increase Resource Revenue. In *IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, pages 267–276, May 2008.

[SS12] Seokho Son and Kwang Mong Sim. A Price- and-Time-Slot-Negotiation Mechanism for Cloud Service Reservations. *IEEE Transactions on Systems, Man, and Cybernetics*, 42(3):713–728, Jun. 2012.

[SSQ+10] D. Saure, A. Sheopuri, Huiming Qu, H. Jamjoom, and A. Zeevi. Time-of-Use Pricing Policies for Offering Cloud Computing as a Service. In *IEEE International Conference on Service Operations and Logistics and Informatics (SOLI)*, pages 300–305, Jul. 2010.

[Sta09] V. Stantchev. Performance evaluation of cloud computing offerings. In *Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP '09. Third International Conference on*, pages 187–192, 2009.

[Str] Stream: synthetic benchmark that measures sustainable memory bandwidth. http://www.cs.virginia.edu/stream/ref.html, Accessed: June 2014,.

[SZXW13] Mingrui Sun, Tianyi Zang, Xiaofei Xu, and Rongjie Wang. Consumer-centered cloud services selection using ahp. *Service Sciences, International Conference on*, 0:1–6, 2013.

[THW02] H. Topcuoglu, S. Hariri, and Min-You Wu. Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. *IEEE*

Transactions on Parallel and Distributed Systems, 13(3):260–274, Mar. 2002.

[TIO] Threaded i/o tester: benchmark to evaluate the performance of the hard disk drive and the file-system. http://sourceforge.net/projects/tiobench/, Accessed: June 2014,.

[TMMVL12] Johan Tordsson, Rubén S. Montero, Rafael Moreno-Vozmediano, and Ignacio Martín Llorente. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future Generation Comp. Syst.*, 28(2):358–367, 2012.

[VCB08] S. Venugopal, Xingchen Chu, and R. Buyya. A Negotiation Mechanism for Advance Resource Reservations Using the Alternate Offers Protocol. In *International Workshop on Quality of Service (IWQoS)*, pages 40–49, Jun. 2008.

[VdBVB10] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove. Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 228–235, 2010.

[VJD+11] Jens-Sönke Vöckler, Gideon Juve, Ewa Deelman, Mats Rynge, and Bruce Berriman. Experiences using cloud computing for a scientific workflow application. In *Proceedings of the 2Nd International Workshop on Scientific Cloud Computing*, ScienceCloud '11, pages 15–24, New York, NY, USA, 2011. ACM.

[VRMB11] Luis M. Vaquero, Luis Rodero-Merino, and Rajkumar Buyya. Dynamically scaling applications in the cloud. *SIGCOMM Comput. Commun. Rev.*, 41(1):45–52, January 2011.

[WABS09] Christof Weinhardt, Arun Anandasivam, Benjamin Blau, and Jochen Stösser. Business models in the service world. In *IEEE IT Pro*, volume 11, pages 28–33, 2009. ISSN: 1520-9202.

[WCC12] Wenting Wang, Haopeng Chen, and Xi Chen. An availability-aware virtual machine placement approach for dynamic scaling of cloud applications. In *Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC), 2012 9th International Conference on*, pages 509–516, Sept 2012.

[xsl] xsltproc: command line xslt processor. http://xmlsoft.org/XSLT/xsltproc.html, Accessed: June 2014,.

[YF12] L. Yazdanov and C. Fetzer. Vertical scaling for prioritized vms provisioning. In *Cloud and Green Computing (CGC), 2012 Second International Conference on*, pages 118–125, Nov 2012.

[YIEO09] N. Yigitbasi, A. Iosup, D. Epema, and S. Ostermann. C-meter: A framework for performance analysis of computing clouds. In *Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on*, pages 472–477, 2009.

[Yue91] Minyi Yue. A Simple Proof of the Inequality FFD(L) $\leq$ 11/9 OPT(L) + 1, $\forall L$ for the FFD bin-packing algorithm. *Acta Mathematicae Applicatae Sinica*, 7(4):321–331, 1991.

[YVCB10] Chee Shin Yeo, Srikumar Venugopal, Xingchen Chu, and Rajkumar Buyya. Autonomic Metered Pricing for a Utility Computing Service. *Future Generation Computer Systems*, 26(8):1368–1380, Oct. 2010.

[zip] 7zip: application to compress files. http://mattmahoney.net/dc/text.html#1789, Accessed: June 2014,.